

Assisted Interpretation of Geological Faults using Machine Learning and Cloud Computing: A Case Study in Pre-Salt environment

Alexandre Augusto Cardoso da Silva*¹, Mario Paes de Almeida Junior¹, Kledson Tomaso Pereira de Lima¹, Pedro Barros Cotta Pesce¹, Caroline Martinez², Gabriella Martins Baptista de Oliveira², Rodrigo Nunes Ferreira² and José Fernando Caparica Junior², ¹Petrobras, ²Schlumberger

Copyright 2021, SBGf - Sociedade Brasileira de Geofísica

This paper was prepared for presentation during the 17th International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 8-11 November 2021.

Contents of this paper were reviewed by the Technical Committee of the 17th International Congress of the Brazilian Geophysical Society and do not necessarily represent any position of the SBGf, its officers or members. Electronic reproduction or storage of any part of this paper for commercial purposes without the written consent of the Brazilian Geophysical Society is prohibited.

Abstract

Assisted interpretation via calibration of models that use machine learning (ML) concepts, through the application of the convolutional neural network (CNN) technique, can significantly reduce the time spent by interpreters with faults/fractures interpretation in seismic data. The fact that model training in the method used requires relatively few labeled data is a major advantage and allows for a clear reduction of time in the interpretation step, mitigating interpreter bias. Furthermore, training of such models in a cloud environment ensures the necessary resources that might, otherwise, lack locally for testing, leading to excessive time spent in this training step. Still, training the model requires care in the labeling of the seismic data, keeping the consistency in the interpretation process. Eventually, the calibration process may require several trained models even for the same seismic data, which can be trained with labeled data interpreted at certain depths or characterized by different structural styles within the same exploratory area. Using this approach, we consider that the time reduction allowed us to finish the fault interpretation phase in 2 weeks, which, otherwise, would take approximately 1-2 months.

Introduction

Broadly speaking, ML technology has been applied in different areas of activity and is reaching different parts of the oil industry. For the model creation process related to geoscience and subsurface, the time reduction in the execution of each step of the modeling workflows adds a significant value, shortening the exploitation cycle time of oilfields. The present work shows the use of machine learning to reduce the interpretation time of geological faults from seismic data, an activity that consumes a considerable amount of the interpreter's time, which is not characterized by a simple mechanical activity. It requires prior knowledge of the structural style associated with the interpreted area, requiring expertise from the interpreter to properly deal with the limitations that the seismic data normally imposes in terms of resolution and quality. The CNN architecture applied here is a variation of U-Net (Figure 1), developed in 2015 with an initial application

focused on segmentation in medical images (Ronneberger et al., 2015) and image processing and pattern recognition (Badrinarayanan et al., 2017).

The recent success of ML approaches has brought new insights into the domain of seismic interpretation, capturing more structural features, in an optimum time frame.

This workflow is based on the synergy of the interpreter's expertise, who provides interpretation labels to train the ML model and uses the advanced image recognition technology (CNN) to estimate a fault prediction volume. This volume is then used in the extraction of faults objects directly from seismic data.

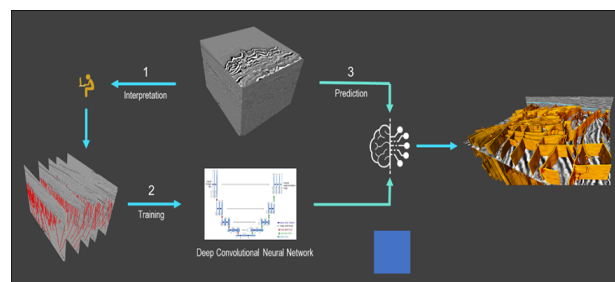


Figure 1 – CNN model training scheme for detecting faults in seismic images. Data courtesy of Geoscience Australia.

The main input data in the evaluation of the ML algorithm was the seismic data of a Pre-salt field. This field is characterized by deepwater oil and natural gas fields, located at a water column of over 2,000m in the Pre-salt Santos basin, offshore of Brazil. The seismic data was acquired throughout a speculative streamer acquisition, presented by WesternGeco, the seismic acquisition branch of the Schlumberger company.

Fault labels were interpreted along inlines and crosslines selected by the interpreter so that the ML algorithm could identify similar fault features in the seismic image.

The seismic volume was not completely interpreted since we focused on the part with greater geological interest, and we also avoided a section of missing seismic data due to the presence of obstructions during the acquisition (Figure 2).

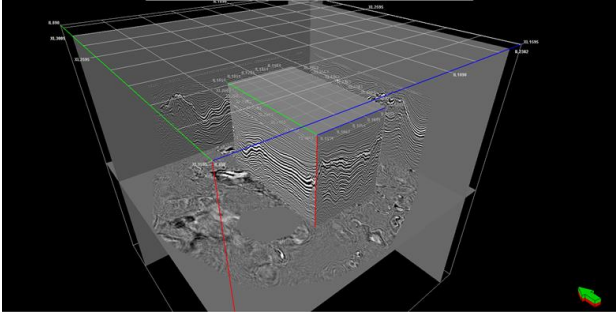


Figure 2 – Detail of the seismic cube interpreted enhancing the area of interest (probe inserted in the larger area).

Method

There were two methods available to perform the training of the CNN algorithm: one method uses the interpretation done on public released seismic volumes to train the CNN algorithm. This database is based on faults/fractures interpreted on various public seismic data, which, therefore, is not necessarily related to the structural style present in our area. On the other hand, this approach may give a first idea of the structural features without requiring a previous interpretation for the labeling process and yielding results that could work as a starting point for the interpretation. In the other method, the training step is based on interpretations performed on the target seismic data itself. Compared to the total size of the volume to be processed by the CNN algorithm, the proportion of the volume to be interpreted and used for labeling in the training and validation steps is equivalent to 1 to 1.5% of the entire volume. This set of 1.5% is still divided into 2 sets: one for training equivalent to 80% and another for validation (20%), using the latter to verify the accuracy of the result obtained by the trained model with the labeled data.

The detection of faults by the CNN was only tested with the original seismic cube; that is, there were no tests involving attributes derived from seismic (attributes that highlight discontinuities, for example). The use of more than one attribute simultaneously was not available during this project. In this sense, for example, the use of the algorithm with multi-azimuthal data can be adapted via solutions that go through stacking the acquired data selecting specific azimuth sections. Also, stacking all azimuth cubes together but weighting them differently, according to the azimuth, may eventually, highlight faults that can be better illuminated by a specific azimuth. The use of the PS wave field was not tested, although the lower resolution, normally observed when compared to the PP wave field, may limit the quality of the interpreted faults.

The faults/fractures used in the CNN training do not need to follow a specific fault structure, that is, there is no need for a fault to be interpreted within its full extent. The suggested criteria that should be followed are: 1- Interpretations should be conducted along inlines and crosslines; 2- Interpretations must be consistent in terms of patterns (for instance, always interpret faults, if

possible, with similar throw in term of visibility, structures with alike dips, fault planes with similar clarity/visibility in the seismic data, etc...), 3- Fault at the edges of seismic cubes and in deeper regions of the data should not be neglected, if they are consistent with the specified pattern and seismic quality.

Also, two constraints are available and can help the ML assisted interpretation process and, possibly, control both the preferential azimuth of the identified faults (Azimuth; not used in this work) and the segmentation of the fault planes into different fault sections, which is controlled at their crossing points (Planarity, used in this work). The automatic fault detection using this ML technique differs fundamentally from other assisted interpretation techniques (e.g., Ant Tracking) and comparisons among them can be difficult to evaluate.

Regarding the consistency in the labeled interpretations, it should be observed that both the training and validation must keep the interpretation pattern. Thus, we avoid inconsistencies between the data used in the training of the model and those used in its validation. Several versions of fault predictions trained with different labels were analyzed and compared carefully. The best model, considering the overall error calculated using a minimization function, is chosen. Inconsistent validation dataset can lead to erroneous conclusions about the accuracy of the final model.

Results

To evaluate the effect of different interpretations (labeled data) on the results derived from the application of the ML algorithm, three tests were conducted with different sets of labels provided as input to the ML algorithm. These input data were interpreted by different interpreters within the project and the final dataset was derived from the union of these interpretations. To maintain the consistency among the interpretations, only interpreted faults showing a high degree of similarity in the seismic image were selected. Eleven crosslines and 3 inlines were interpreted, observing the need to have from 1% to 1.5% of the entire volume interpreted as previously mentioned. The labeled lines were divided into 80% for training (9 crosslines and 2 inlines) and 20% for validation (2 crosslines and 1 inline).

The input datasets selected for the tests were elaborated seeking the evaluation of specific scenarios. The input datasets are described below:

- First dataset: only faults interpreted in the Pre-Salt were included (run1);
- Second dataset: Pre-Salt faults and discontinuities features observed in the salt layer were included (run2);
- Third dataset: only faults interpreted in the Pre-Salt, but with small differences in the dip of some faults compared to the interpretation of the first dataset (run3).

Comparing the results of the three scenarios, it was possible to evaluate how the input datasets changed the results for each trained model. Each model took

approximately 3 hours to be trained. Next, we show the comparisons, starting with the first scenario (run1).

Figures 3 and 4 show a composition of crossline A (figure 3-C), one of the interpreted crosslines, and crossline B (figure 4-C), which was not interpreted. Both show the result of automatic interpretation via ML (in purple). The manual interpretation, provided for model training (in orange), is shown overlapping the automatic interpretation in the crossline A. Besides the seismic sections, maps show all crosslines (12) and inlines (3), which were manually interpreted and provided for training and validation (figures 3-A and 4-A), and all faults automatically interpreted (figures 3-B and 4-B, in purple).

It is possible to see that structures/faults mapped via ML largely follow the faults provided as input in the labeled dataset. This is clearer viewed in the crossline B. It was not interpreted manually, but the ML algorithm was able to identify faults manually mapped in other crosslines also present in the crossline B. However, some faults which were, apparently, similar to faults previously mapped during the labeling phase, were not identified by the calibrated ML model. This may indicate the necessity to include more labeled faults for the ML algorithm.

Figures 5 and 6 show the comparison of the second scenario (run2) with the automatic interpretation in blue and the manual interpretation in red, in a similar fashion shown in figures 3 and 4 for the first scenario (run1). Note that for this scenario the labeled dataset included features that resemble faults in the salt strata (figure 5-C). Moreover, other Pre-Salt faults were included, sometimes showing slight differences in the dip of the faults previously provided. What is perceived is that the pattern of faults identified by the algorithm changes substantially, with the detection of new faults, which are not so clear in the salt strata. There is also a change in the pattern of the faults automatically detected in the Pre-Salt when compared to the first scenario (run1), even though most of the faults used as input for training the model of the first scenario are also present in the second (run2).

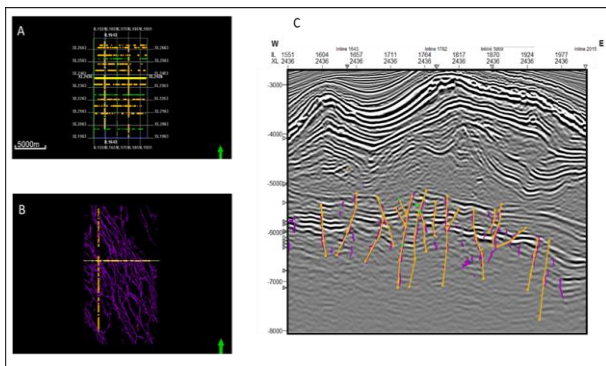


Figure 3 – Composition showing the crossline A (c) overlapped by manual interpretation (orange) and by the result of automatic detection via ML (purple). Besides the seismic section, it is shown the map (A) with the manually interpreted lines (labeled data) and the result of automatic detection via ML (B) throughout the analyzed area.

These differences observed between the first and second scenarios help to highlight the sensitivity of the ML based interpretation algorithm to the pattern of the labeled faults provided as the input to the training process. It is possible that even faults manually interpreted outside of the area of interest—the Pre-Salt—for example, can influence the training process of the model and can impact the behavior of the trained model. This experiment leads us to believe that the interpretations provided for the model training step should keep a systematic and coherent pattern, possibly working with different training datasets for models focused on different structural styles. For example, ML models that focus on Pre-Salt should have within the input training dataset only faults interpreted in this environment. Or, in case the objective is in the Post-Salt region, only faults interpreted in this region should be provided during the training step of the model.

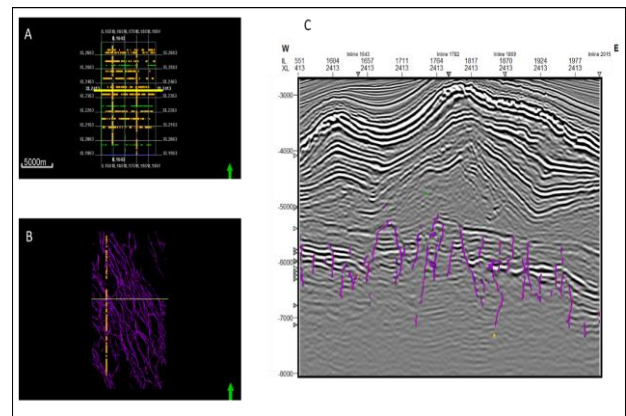


Figure 4 – Composition showing the crossline B (c) overlapped by the result of automatic detection via ML (purple). Besides the seismic section, it is shown the map (A) with the manually interpreted lines (labeled data) and the result of automatic detection via ML (B) throughout the analyzed area.

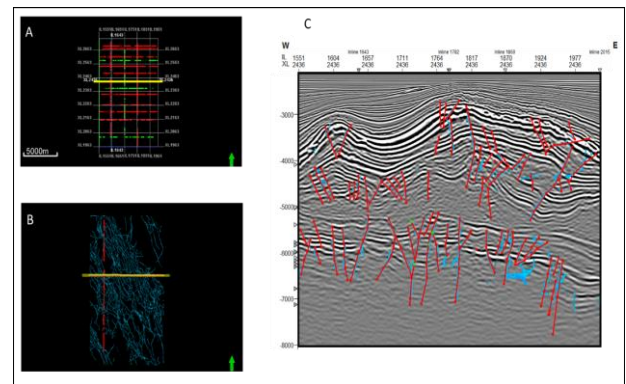


Figure 5 – Composition showing the crossline A (c) overlapped by manual interpretation (red) and by the result of automatic detection via ML (blue). Besides the seismic section, it is shown the map (A) with the manually interpreted lines (labeled data) and the result of automatic detection via ML (B) throughout the analyzed area.

Figures 7 and 8 show the results of the third scenario (run3). This scenario used as the input dataset interpreted faults (labeled) that are very similar to those belonging to the training dataset used in the first scenario (run1), only with the inclusion of faults with small differences in the dip plus the inclusion of a reduced number of faults with small lengths. Note that even these small differences are already enough to cause differences in the automatic interpretation using the ML algorithm. Again, this seems to be evidence of the sensitivity of the trained model to the labeled dataset provided in the training/validation steps.

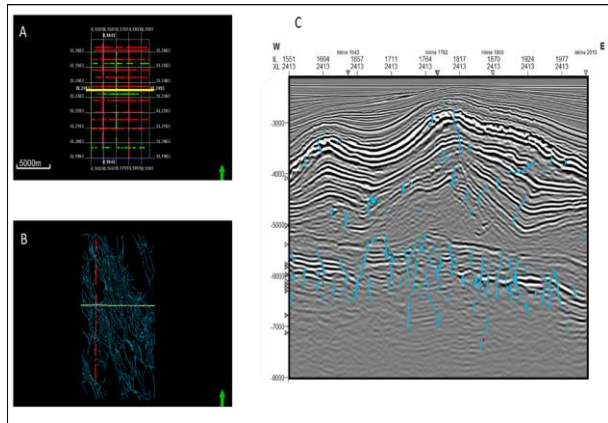


Figure 6 – Composition showing the crossline B (c) overlapped by the result of automatic detection via ML (blue). Besides the seismic section, it is shown the map (A) with the manually interpreted lines (labeled data) and the result of automatic detection via ML (B) throughout the analyzed area.

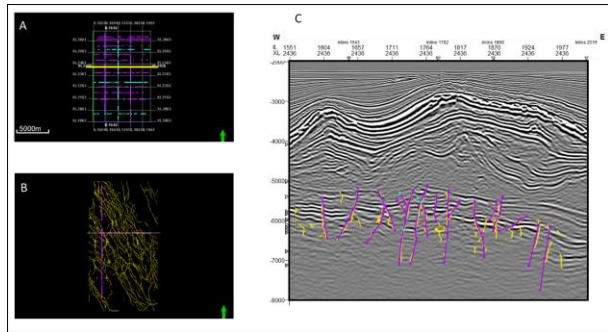


Figure 7 – Composition showing the crossline A (c) overlapped by manual interpretation (purple) and by the result of automatic detection via ML (yellow). Besides the seismic section, it is shown the map (A) with the manually interpreted lines (labeled data) and the result of automatic detection via ML (B) throughout the analyzed area.

Fault Extraction: calculation of Planarity and Azimuth cubes

After the automatic interpretation step using the trained models, the obtained faults/structures went through a process called *Fault Extraction* to transform the identified features by the ML algorithm into *point sets*, which can be

individualized into faults and edited later. The first phase of *Fault Extraction* is associated with two auxiliary properties used, respectively, in the separation of intersecting faults (Planarity) and sub-division based on azimuth sectors, according to an azimuth criterion (*Azimuth*).

In this phase, two parameters—minimum threshold (varying from zero to one) and radius (defined in terms of voxels number)—are defined so that the faults detected with the ML algorithm can be extracted. The former parameter is responsible to inform the extraction process of what can be considered a fault or only a background noise and the latter defines a search radius (usually twice the thickness of the faults). A geometrical analysis is undertaken using a principal component analysis to determine Planarity and Azimuth attributes (Etchebes and al., 2019).

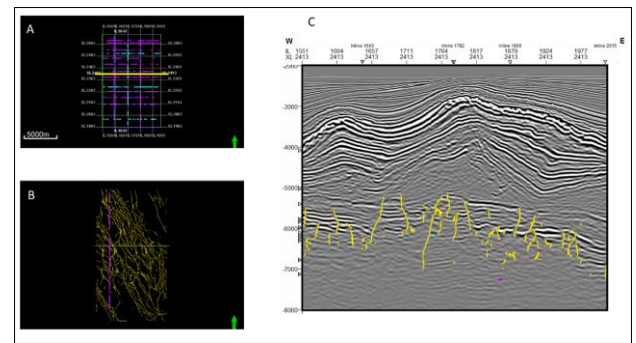


Figure 8 – Composition showing the crossline B (c) overlapped by the result of automatic detection via ML (yellow). Besides the seismic section, it is shown the map (A) with the manually interpreted lines (labeled data) and the result of automatic detection via ML (B) throughout the analyzed area.

The fault extraction process was applied both to the first (run1) and third scenario (run3) identified faults. The values used for the extraction were equal to 0.2 and 4 for the minimum threshold and radius, respectively, in both scenarios. Figure 9 shows, on a map extracted at depth 5,830 ms the results of Planarity and Azimuth with the parameters above, calculated using the third scenario identified faults as the input dataset. Note that the faults identified using the ML technique show a structural pattern with faults NW oriented, which is characteristic of the studied area.

The *point sets* representing each fault plane were obtained after the editing process using the *Planarity* propriety. In this work, the *Azimuth* propriety was not used for extraction purposes since the faults identified using the ML algorithm already follow the general orientation of the geologic structures of the area.

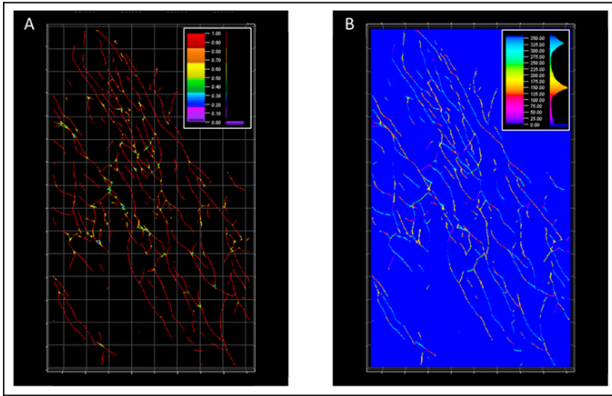


Figure 9 – Results for the Planarity (A) and the Azimuth (B) obtained from fault extraction of the scenario three (run3). Maps extracted at depth of 5,830 ms.

The *point sets* went through a manual quality control process that aimed to combine different *point sets* as a single fault, if necessary. *Point sets* that do not represent faults were eliminated and regions with crossing faults were corrected. Although this final quality control is done manually, the time spent for this step is reduced, given the presence of the previous controls during the fault extraction phase and the calculation of the *point sets*. **Figure 10** shows the *point sets* after manual quality control.

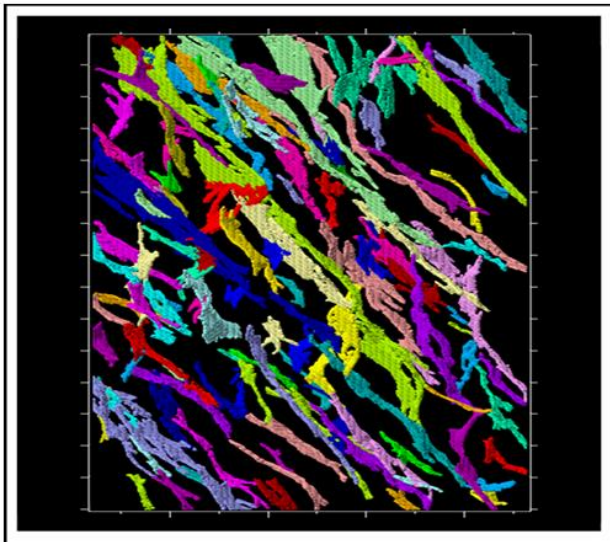


Figure 10 – Point sets extracted during the fault extraction phase after the quality control and manual editing. The point sets shown were obtained using the output from the scenario three (run3).

Quality control via stereograms: ML vs. manual interpretation

One way to compare the result of interpretation using ML methods with manual interpretation is to use stereograms. Thus, we can see if the average behavior of azimuth and dip of the faults obtained via the two ways are similar. The

fault poles were calculated based on the *point sets* of the faults detected after the manual editing process.

Figures 11 and 12 show the stereograms for the *point sets* after editing (figure 11) and for the *point sets* after conversion to surfaces (figure 12). The figures help to identify potential issues detected in the first (11-A and 12-A) and the third scenarios (11-B and 12-B).

It is visible in the stereograms of *point sets* before conversion to surfaces (figure 11) that there are few differences between the results of the first (A) and third (B) scenarios. Such differences are derived from the small changes in the faults labeled for the training of each model. Analyzing the stereograms (figure 12) after the conversion of *points sets* to surfaces, there is a slight increase in the dip of the faults in both scenarios. This effect is derived from the smoothing process imposed on the surfaces to better adapt the surfaces to the *points set*.

Figure 13 shows the stereograms of the manually interpreted faults (color density) and the poles derived from the *point sets* of the first scenario (run1), that is, before their conversion to the surfaces. Note that the stereogram showing the manually interpreted faults (figure 13-A) shows azimuths and dips similar to those observed in the stereogram representing the faults detection via ML (figure 13-B).

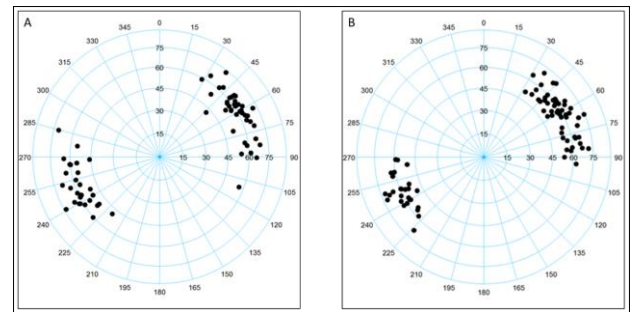


Figure 11 – Stereograms with faults detected in the first (A) and third scenarios (B). The poles shown represent the *point sets* after quality control but before conversion to surfaces.

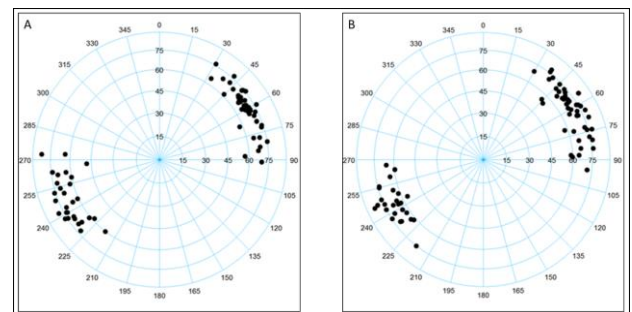


Figure 12 – Stereograms with faults detected in the first (A) and third scenarios (B). The poles shown represent the *point sets* after conversion to surfaces.

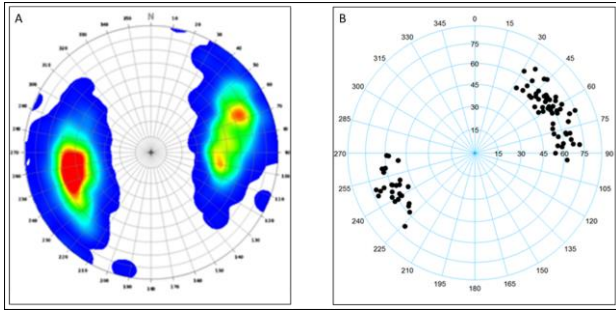


Figure 13 – Stereograms comparing the result of manual interpretation (A) and automatic detection (B) obtained from the first scenario (run1).

Conclusions

The assisted interpretation via model calibration, using ML concepts through the application of the CNN technique, reduces importantly the time spent by interpreters to interpret faults. The fact that the training of the models requires relatively a small amount of labeled data is an important advantage, reducing the time in the interpretation stage. In this work, we consider that the time reduction allowed us to finish the fault interpretation phase in 2 weeks, which otherwise, would take approximately 1-2 months.

Another advantage is the training model step being executed in a cloud environment, ensuring the required resources for both the model training step and parameter testing. Local machines/servers may lack the computational resources needed to conclude the processing in a feasible time frame.

The approach, although machine-assisted, still requires the expertise of the geoscientist. We can mention that the need for consistency of labeled input datasets for training the models is critical. In this sense, the consistency among the interpreted faults of the labeled dataset — both for training and validation of the model — should always be observed. Furthermore, it is worth noting that different structural styles, for instance, the Pre-Salt and Post-Salt, will probably require separately trained models, increasing the time spent in the training phase.

Acknowledgments

The authors would like to acknowledge the effort and support of Petrobras and Schlumberger throughout the duration of this project, ensuring the resources and staff needed to conclude it. We are also thankful to WesternGeco Multiclient for the approval to publish the results reached in this project.

References

RONNEBERGER, Olaf. FISCHER, Philipp. BROX, Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. <https://arxiv.org/pdf/1505.04597.pdf>. Disponível em: 16/12/2020.

BADRINARAYANAN, V., A., KENDALL, and R., CIPOLLA, 2017, Segnet: A deep convolutional encoder-decoder architecture for image segmentation: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. 39, 2481–2495, doi: <https://doi.org/10.1109/TPAMI.2016.2644615>.

M. ETCHEBES, T. H. BOE, A. BOUNAIM, T. BRENNAN, D. STECKHAN, M. NICKELN, H. G. BORGOS, P. LE GUERN, W. ATHMER, B. H. FOTLAND, K. S. BOGE. Innovative workflow for structural seismic interpretation: Combining automated technologies with structural analysis techniques. SEG International Exposition and 89th Annual Meeting. 2019. <https://library.seg.org/doi/10.1190/segam2019-3215061.1>