PUC

ISSN 0103-9741

Monografias em Ciência da Computação

n° 03/2018

Seismic Fault Detection Using Convolutional Neural Networks trained on Synthetic Post-stacked Amplitude Maps

Axelle Pochet

Pedro Henrique Bandeira Diniz

Hélio Lopes

Marcelo Gattass

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 03/2018 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena July, 2018

Seismic Fault Detection Using Convolutional Neural Networks trained on Synthetic Post-stacked Amplitude Maps *

Axelle Pochet[1] , Pedro Henrique Bandeira Diniz,
Hélio Lopes, Marcelo Gattass[1]

[1]Instituto Tecgraf – PUC Rio

axelle@tecgraf.puc-rio.br, pbandeira@inf.puc-rio.br,
lopes@inf.puc-rio.br, mgattass@tecgraf.puc-rio.br

Abstract. Fault detection is a crucial step in reservoir characterization. Despite the many tools developed in the past decades, automation of this task remains a challenge. We investigate the application of Convolutional Neural Networks (CNN) to seismic fault detection. CNN is a deep learning method growing in interest in the computer vision community, due to its high performances in a great variety of object detection tasks. One of the constraints of this method is the need to provide a massive number of interpreted data, a requirement particularly difficult to attend in the seismic area. To this end, we built a synthetic dataset with simple fault geometries. The input of our network is the seismic amplitude only; the method does not require computing any seismic attribute. We apply a strategy of patch classification along the images, which requires a simple post process to extract the exact fault location. Our network shows good results on synthetic data and encouraging results when tested on regions of a real section of the Netherland offshore F3 block in the North Sea.

Keywords: Convolutional Neural Networks, Hough Transform, Seismic Fault.

Resumo. Detecção de falhas é um passo crucial para a caracterização de reservatórios. Apesar das diversas ferramentas desenvolvidas nas últimas décadas, a automação desta tarefa continua sendo um desafio. Investigamos a aplicação de Redes Neurais Convolucionais (RNC, do inglês Convolutional Neural Networks ou CNN) para detecção de falhas sísmicas. RNC é um método de aprendizagem profunda com crescente interesse na comunidade de visão computacional, devido a sua alta performance em uma grande variedade de tarefas de detecção de objetos. Uma das restrições deste método é a necessidade de uma grande quantidade dados interpretados, um requisito particularmente difícil de atingir na área da sísmica. Para este propósito, foi construído um conjunto de dados sintéticos com geometrias de falha simples. A entrada da rede proposta é apenas a amplitude sísmica; o método não exige o cálculo de nenhum atributo sísmico. Aplicamos uma estratégia de classificação de *patches* ao longo das imagens, o que requer um pós-processamento simples para extrair a localização exata da falha. Nossa rede mostra bons resultados nos dados sintéticos gerados e resultados encorajadores quando testados em regiões de uma seção real do bloco F3 marítimo da Holanda no Mar do Norte.

Palavras-chave: Redes Neurais Convolucionais, Transformada de Hough, Falhas Sísmicas.

* This work has been sponsored by Shell Brazil. It has been submitted to IEEE Geoscience and Remote Sensing Letters (GRSL) journal.

In charge of publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

1 Introduction

The past decades have seen the development of many tools for computer-aided fault detection. The vast majority of these methods are based on the use of seismic attributes. Those measurements, usually made in the post-stack stage, allow enhancing possible fault location by looking at the local continuity of the seismic signal (coherence [Bahorich and Farmer, 1995][Luo et al, 1996], semblance [Marfurt et al, 1998], variance [Van Bemmel and Pepper, 2000], chaos [Randen et al, 2001], edge detection [Di and Gao, 2014]), or at the geometry of the reflectors (curvature [Lisle, 1994][Roberts, 2001][Al-Dossary and Marfurt, 2006], flexure [Gao, 2013]). An alternative is to use the information of interpreted horizons to find fault locations (horizons dip and azimuth maps, [Rijks and Jauffred, 1991]).

Each seismic attribute has its pros and cons, and fails at enhancing faults only; numerous artifacts remain, other structures appear. Seismic attributes usually require massive computation, and, alone, are not suited for efficient fault identification: a human interpreter must spend time finalizing the study manually. Consequently, many authors propose to post-process the attribute maps to extract fault location automatically. For example, [Gibson et al, 2005] use semblance to create a set of high faultiness points that are joined to build fault surfaces; [Zhang et al, 2014] apply a skeletonization on the coherence cube to extract fault sticks; and [Wang and AlRegib, 2014][Wang and AlRegib, 2017] use the Hough Transform to extract fault locations from binarized continuity maps.

Another way to use attribute information is to combine them. Machine Learning algorithms are particularly suited for this task, as they can efficiently find relationships between a set of input features (seismic attributes) and the desired output value (fault location). The work of [Tingdahl and de Rooij, 2005] uses a neural network to combine a set of 12 attributes and generate a fault probability map. More recently, [Di et al, 2017] combined 14 attributes in a multi-attribute Support Vector Machine, a powerful supervised learning technique. Such methods give accurate results but are still computationally expensive: in addition to the set of attributes that must be computed for each new point classification, supervised techniques need as input a large amount of interpreted data that can be consequent depending on the algorithm applied.

The need for a large amount of interpreted data could explain why more powerful algorithms such as deep learning methods are still scarce in the seismic literature. Such algorithms can automatically extract meaningful information from the original amplitudes through their combination in the network's hidden layers, creating new features dynamically during training. Despite the usually substantial time to train those networks, once the training is done, further outputs are obtained very efficiently. There are different approaches to apply such methods to the problem of fault detection in seismic data. [Di et al, 2018] created a dataset by manually labeling crosslines from a seismic cube, then achieved fault detection in other sections of the same cube. [Araya-Paulo et al, 2017] trained deep networks with thousands of synthetic volumetric velocity models, obtained by approximating acoustic wave equation to generate wave fields as time-series signals with predefined acquisition geometry. [Huang et al, 2017] proposed to train their networks on top of several seismic attributes, using synthetic seismic cubes with simple fault configurations. Despite achieving good results, the authors trained and classified data in the same field, making it difficult to judge the generalization capability (and thus the practical usability) of the proposed methods.

In this work, we train Convolutional Neural Networks (CNNs) with synthetic data and try to apply the classifiers on real data to study the generalizability of the model. Designing a robust, general classifier would indeed save the heavy step of training and tuning a CNN for each new dataset. CNNs combine a series of convolution steps with a fully connected neural network to perform classification, and recently proved to be powerful in various computer vision tasks ([Rawat and Wang, 2017], [Zhiqiang and Jun, 2017]), among which seismic objects detection ([Waldeland and Solberg, 2017],[Di et al, 2018],[Huang et al, 2017]). Compared to real data, synthetic data present the great advantage to provide total control on the ground truth and are easily scalable in terms of the number of inputs. Our input is the seismic amplitude only, and our method does not require any prior seismic attribute computation. We use a patch classification scheme on synthetic data for network training, as explained in the next section. Section 3 shows the results of our implementation of the proposed method on synthetic and real data, and in section 4, we draw some conclusions.

2 Method

We propose a methodology in four steps. First, we generate synthetic seismic images where we control the location of the faults. Second, we extract Fault and Non-fault patches from the generated dataset. Then, we train and fine-tune different CNN architectures focusing on maximizing quality metrics. Finally, we classify pixels in new images (synthetic and real) and post process the results for fault segmentation.

2.1 Synthetic dataset generation

The difficulty in obtaining many good quality fault interpretations on real seismic images led us to investigate the use of synthetic data with known fault positions. The open source code IPF from Dave Hale [Hale, 2014] allowed us to reproduce the results of migrated, post-stacked seismic data. Beginning with a randomly generated reflectivity model extended along the section, simple image transformations recreate sequential rock deformations along time: shearing, folding and faulting. We can then apply convolution with a Ricker wavelet and add random noise. Each step of the process can be parameterized. We built a dataset of 500 images of 572x572 pixels, all containing one straight fault crossing the section entirely, modifying randomly fault angle, position and throw, shearing slope, folding amplitude and frequency, wavelet peak and amount of noise. Resulting images present amplitude values between -1 and 1. Along with the seismic amplitude information, we generated for each image its corresponding binary mask, indicating in white the location of the fault. Fig. 1 shows an example of such a pair.

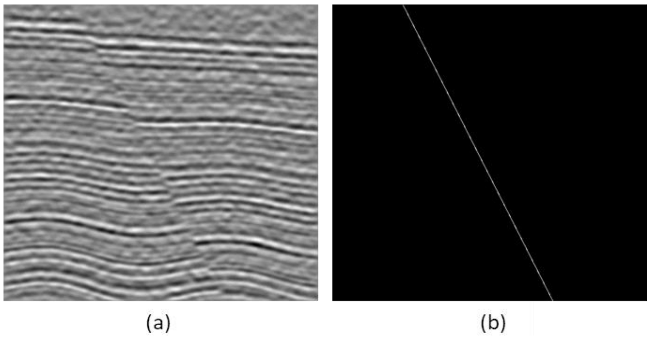

Fig. 1. Synthetic Seismic Image. (a) Example of a synthetic seismic image from our dataset (b) The corresponding binary mask.

2.2 Patches Extraction

Many machine learning techniques use features extracted from images as input. Features are relevant information that we think could be efficiently combined to achieve the desired classification. One of the advantages of CNN is that it does not require an explicit feature extraction step. Instead, the neural network uses the image itself as input and attempts to extract the best features implicitly. Applying these principles to the seismic imagery area, good feature candidates are naturally any fault enhancing seismic attribute. Such attributes are computed using a small neighborhood of seismic amplitude values. Seismic amplitude is thus at the core of the fault detection problem, and a small neighborhood of amplitude values can be used as input to a CNN, that will hopefully find and compute the best seismic attributes dynamically, without the need of explicitly passing them as input. This small neighborhood is what we call here a patch.

Since faults may be located anywhere in the seismic image, all pixels are fault candidates. Our approach seeks to classify all pixels as a fault or a non-fault pixel. A patch is composed by the candidate pixel itself at the center and its neighbor pixels. The classification of a pixel is the classification of its patch. To separate the pixels in our two target classes, Fault and Non-fault, we use binary mask images, which contain the marking of the faults. If a pixel in the seismic image is masked by a white pixel in the binary image, this pixel is considered as Fault. Similarly, black pixels in the binary masks are considered Non-fault. Additionally, if a pixel is Non-fault but the fault passes somewhere in its patch (partial faulting), we discard the patch: in this work, such patches are simply not trained. Fig. 2 shows an example of one Fault patch and one Non-fault patch extracted from a synthetic seismic image.

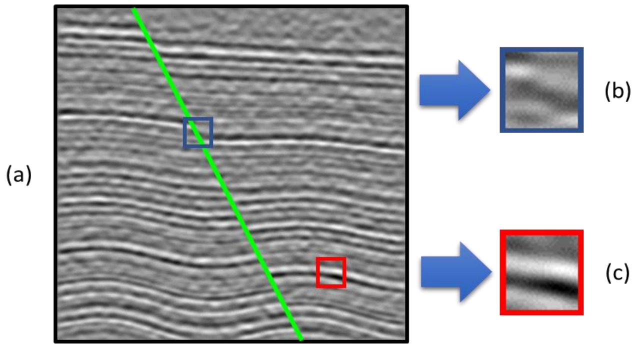

Fig. 2. Extracting patches. (a) Seismic image crossed by a fault. The fault location is highlighted. (b) Fault patch. (c) Non-fault patch.

Follows how the sets of patches are used as input to the CNN:

- For training images, we extract all possible Fault patches and one Non-Fault patch every 23 pixels. This generates a balanced number of patches for the two classes, which is desirable for training.
- For validation images, we extract all possible Fault patches and one Non-Fault patch every 10 pixels, to account for classes' natural imbalance in practice, and thus obtain interpretable quality metrics.
- For test images, we extract one patch every 3 pixels, regardless of the binary mask. This small pixel step ensures the fault will be crossed, while efficiently generating classifications suited for visualization.

2.3 CNN Training

The common architecture of a CNN is described in Fig. 3. The convolutional layers have trainable filters applied throughout the patches. Pooling layers perform non-linear down-sampling. We use max-pooling, which yields maximum values over a neighborhood of feature maps. Activation layers apply non-linear functions on input neurons. Here, we use the rectified linear unit (ReLU), since it provides several times faster training than other activation functions [Krizhesky et al, 2012], usually avoids the vanishing gradient problem and promotes model sparsity [Glorot et al, 2011]. The last part of the network is a set of fully-connected layers: the number of input neurons is defined by the number of pixels resulting from the previous layer. Dropout layers prevent the generation of overfitted models, which are common in complex networks. They are generally placed deep in the network, after layers learning a large set of parameters, since this is where the overfitting risk is high. Supervised training is carried out using a form of stochastic gradient descent (SGD) to minimize the discrepancy between the desired output and the current output of the network, based on some loss function [Lecun et al, 1998]. We choose the Softmax loss function in the last layer, which outputs a normalized probability for each class. The final classification is the class with the highest score.

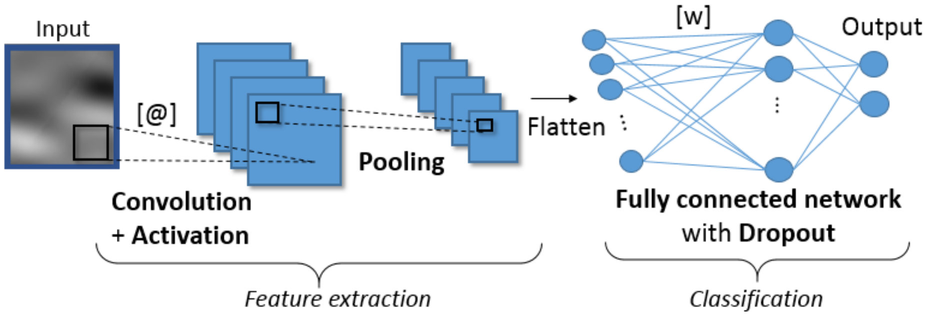

Fig. 3. Architecture of a CNN, with: [@] = trainable filters; [w] = trainable weights. Activation and dropout are optional layers. Flatten operation converts the 2D list of matrices into a 1D list of neurons. The Feature extraction step can contain any number of Convolution and Pooling layers.

We applied and tested different CNN architectures and hyper-parameters, along with the input patch size and resolution. Beginning with a common LeNet [Lecun et al, 1998], we added complexity, applying results from the VGG-Net [Simonyan and Zisserman, 2014], since deeper networks are better at differentiating classes. Another aspect is that with deeper networks, small convolution masks of 3x3 pixels should give good results. However, adding too many layers to a network can have negative effects: the training time increases dramatically, and the classification can fall into overfitting the training data. Consequently, there is a trade-off to find. All training sessions, additionally, shared some parameters: we used a learning rate value of 0.001, momentum value of 0.9 and input batch of 30 patches. We used 400 seismic images for training (381,079 patches), 50 images for validation (148,632 patches) and kept the remaining 50 images to perform tests.

For each configuration, we estimate the quality of the classifier on six common criteria, considering the Fault class as the positive class: accuracy, sensitivity, specificity, F1-score, Area Under the ROC curve (AUC) and a visual evaluation on entire sections of the test set. Sensitivity, the capacity of the network to output true positives, should be high enough to underline the faults. Specificity should be as close to 1 as possible as even a small number of false positives tend to give poor visual results on the test sections.

3 Results

After training several CNN architectures on synthetic data, we find that the smallest patch size that gives satisfying results is 45x45. In this section we present two different CNN architectures, the first giving the highest metrics on the synthetic test set, the second giving the most satisfactory visual result when classifying real data patches

3.1 Results on Synthetic Data

The best network for fault detection on synthetic data was obtained with the architecture described in Table 1. It gave an accuracy of 0.98, a sensitivity of 0.95, a specificity of 0.99, F1-score of 0.97 and AUC of 0.99.

Layers	C	C	C	ReLU	MP	ReLU	C	C	C	ReLU	MP	ReLU	FC	ReLU	Dropout	FC	ReLU
#F	20	20	20				50	50	50								
MS	3x3	3x3	3x3		2x2		3x3	3x3	3x3		2x2						
#N													16			32	

Table 1. CNN Architecture that achieved the best results on synthetic data, after 50 epochs. With: #F = number of filters; MS = mask size; #N = number of neurons; C= Convolution, MP = Max-Pooling, FC = fully connected layer.

Fig. 4 (left column) shows the classification and fault extraction of a section from the test set. Classification is performed every three pixels. The fault is clearly highlighted, but still coarse because patches crossing the fault partially are classified as Fault. Since the training step did not include such patches, as stated in section II.B, those results are unsurprising. Note also that such patches did not enter in the calculation of the quality metrics, over-estimating all values except sensitivity.

We observed similar results on all test sections, including sections with configurations that were not shown during training: varying fault throw (Fig. 4(center column)), faults crossing each other (Fig. 4(right column)). Indeed, as patches contain local image information, a variety of fault geometries can be detected even when not specifically trained.

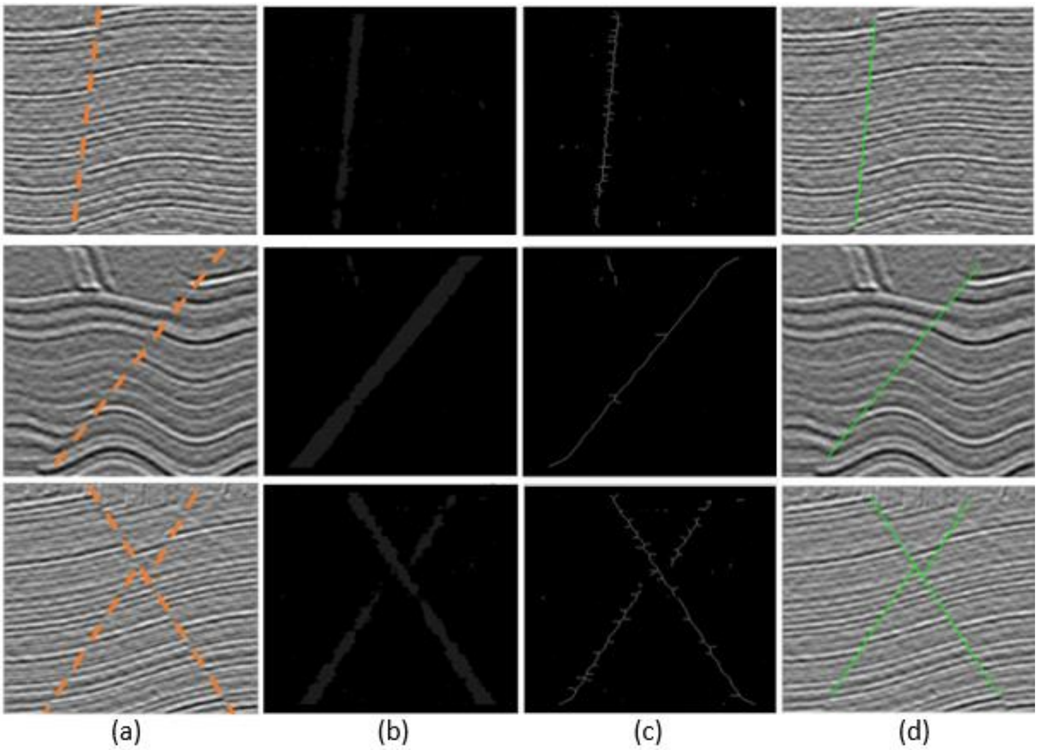

Fig. 4. Classification and fault extraction on synthetic test sections. (a) Input section with expected fault marked in dashed lines. (b) Raw classification of 1 over 3 pixels. (c) Results of erosion, dilation and thinning on (b). (d) Extracted faults using the Hough transform on (c).

3.2 Results on Real Data

We test our CNNs on a real data of the North Sea, the F3 cube [OpenDTect]. The selected section presents different fault regions. Manual interpretation of the faults in each region can be observed in Fig. 5 (a, b, c: top left). To test our classifier, we apply the following process: first, we set the amplitude values of the entire section between -1 and 1, clipping the histogram under a threshold to enhance visualization, in order to set the section as close as possible to the training conditions. Second, we extract one image per region of interest. Third, for each image we extract patches with a visual close to the synthetic patches, because the horizons' scale in the real section is different from the one in the synthetic dataset. Hence, extracting directly 45x45 patches from the real data can lead to poor results. We assume that the horizon's scale should be homogeneous along the regions of interest and thus visually select a single patch size per region. We resize those patches to the expected classification size of 45x45, using a bicubic interpolation. We classify every pixel in the images. The best CNN architecture in this case was chosen looking only at the visual results and is summarized in Table 2.

Layers	C	C	ReLU	MP	ReLU	C	C	ReLU	MP	ReLU	FC	ReLU	Dropout	FC	ReLU
#F	20	20				50	50								
MS	3x3	3x3		2x2		3x3	3x3		2x2						
#N											16			32	

Table 2. CNN Architecture that achieved the best results on synthetic data, after 60 epochs. With: #F = number of filters; MS = mask size; #N = number of neurons; C= Convolution, MP = Max-Pooling, FC = fully connected layer.

Fig. 5(a) shows the detection of a single straight fault. Fig. 5(b) shows several parallel, sub-vertical and well-defined straight faults, detected by extracting the main peaks in the Hough transform process. Fig. 5(c) presents similar results on a noisy, multi-faulted region. These multi-faulted regions, using smaller patches (8x8 resized to 45x45) allowed to separate the faults despite the coarse classification. Additionally, in these regions there was no need to apply morphological operations described in section II.D before applying the Hough Transform.

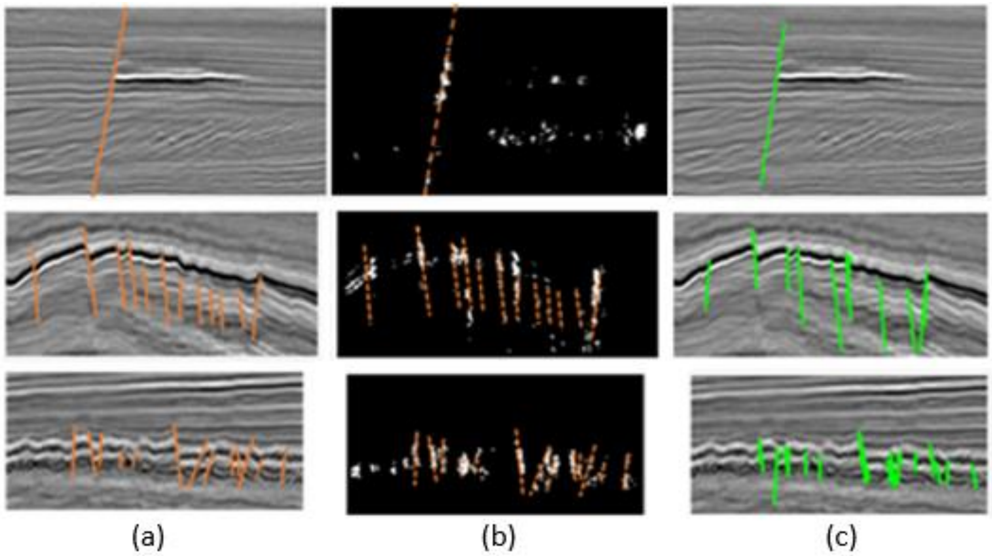

Fig. 5. Classification and segmentation on real test sections. (a) Input section with expected fault location (manual picking). (b) Classification of all pixels, with superposition of picking for better visualization. (c) Extracted faults using our post-processing procedure. Top; Single straight fault; Middle: Several sub-vertical faults; Bottom: Several sub-vertical faults with noise.

When applying this second CNN on the synthetic validation set, we obtain lower quality metrics than the CNN presented in Table 1 (accuracy: 0.94, sensitivity: 0.69, specificity: 0.99, F1-score: 0.80, AUC : 0.96). We assume that despite a loss in quality for Table 2 network when applied to the synthetic data, this CNN has actually a greater generalization capacity.

4 Conclusion

This paper presented a methodology for the detection and segmentation of faults in seismic images, using a patch-based CNN approach for training and classification, and line Hough Transform for post-processing. CNNs were trained with synthetic data only and yielded good results when applied to new synthetic data, with a perfect match of the predicted and ground truth fault after applying the post process. They also revealed promising results when classifying pre-processed real data. Interestingly, the CNN with the highest performance in both synthetic and real case is different: Table 2 CNN revealed a greater generalization capacity than Table 1 CNN, despite poorest quality metrics on the synthetic validation set. The use of CNN allowed us to train with the seismic amplitude map as the only input feature: explicit feature extraction and selection steps were unnecessary. However, we highlighted the large number of empirical parameters used in CNNs, which makes model fine-tuning difficult.

There are a few changes that can be applied to achieve higher performance, especially in real seismic images. First, the patch size in the case of real data could be extracted automatically, considering whether the scale of the horizons (using image texture measurements), or the seismic signal frequency (patches should have a similar frequency range). Ideally, we could try to generate a dataset containing the commonly seen real seismic signal frequencies. Second, the architecture of the CNN could be improved. To do so, we could try to visualize and better understand the generated features, or we could use evolutionary or other optimization algorithms to find the best architecture automatically. The question of patches with partial faulting should also be addressed. Including such patches as a new label in the classification could be a way to refine classification around the fault. Another line of research could be the improvement of our post-processing, following better skeletonization algorithms, as [Wang et al, 2014] for example. Finally, while promising results on real data show the relevance of the synthetic training methodology, it is clear that it may be more efficient to use real seismic images for training. Therefore, we are currently trying fine-tuning and transfer learning methods to include some real data in the training and better adapt our CNN to real data.

Acknowledgements

The authors gratefully acknowledge support from Shell Brazil through the “Coupled Geomechanics” project at Tecgraf Institute (PUC-Rio) and the strategic importance of the support given by ANP "Compromisso com Investimentos em Pesquisa e Desenvolvimento" through the R&D levy regulation.

References

AL-DOSSARY, S. and MARFURT, K. 3D volumetric multispectral estimates of reflector curvature and rotation. Geophysics, vol. 71, no. 5, pp. P41-P51, 2006.

ARAYA-POLO, M., DAHLKE, T., FROGNER, C., ZHANG, C., POGGIO, T. and HOHL, D. Automated fault detection without seismic processing. The Leading Edge, vol. 36, no. 3, pp. 208-214, 2017.

BAHORICH, M. and FARMER, S. 3-D seismic discontinuity for faults and stratigraphic features: The coherence cube. The Leading Edge, vol. 14, no. 10, pp. 1053-1058, 1995.

DI, H. and GAO, D. Gray-level transformation and Canny edge detection for 3D seismic discontinuity enhancement. Computers & Geosciences, vol. 72, pp. 192-200, 2014.

DI, H., SHAFIQ, M. and ALREGIB, G. Seismic-fault detection based on multiattribute support vector machine analysis. SEG Technical Program Expanded Abstracts 2017, 2017.

DI, H., WANG, Z. and ALREGIB, G. Seismic fault detection from post-stack amplitude by convolutional neural networks. 80th EAGE Conference and Exhibition, 2018.

GAO, D. Integrating 3D seismic curvature and curvature gradient attributes for fracture characterization: Methodologies and interpretational implications. Geophysics, vol. 78, no. 2, pp. O21-O31, 2013.

GIBSON, D., SPANN, M., TURNER, J. and WRIGHT, T. Fault surface detection in 3-D seismic data. IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 9, pp. 2094-2102, 2005.

GLOROT, X., BORDES, A. and BENGIO, Y. Deep Sparse Rectifier Neural Networks. Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, 2011.

HALE, D. Seismic image processing for geologic faults. https://github.com/dhale/ipf, 2014.

HOUGH, P. V. C. Method and means for recognizing complex patterns. U.S. Patent No. 3,069,654, 1962.

HUANG, L., DONG, X. and CLEE, T. A scalable deep learning platform for identifying geologic features from seismic attributes. The Leading Edge, vol. 36, no. 3, pp. 249-256, 2017.

KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, pp. 1097-1105, 2012.

LECUN, Y., BOTTOU, L., BENGIO, Y., HAffNER, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE vol. 86, no. 11, pp. 2278-2324, 1998.

LISLE, R.J. Detection of Zones of Abnormal Strains in Structures Using Gaussian Curvature Analysis. *AAPG Bulletin,* vol. 78, 1994.

LUO, Y., HIGGS, W. and KOWALIK, W. Edge detection and stratigraphic analysis using 3D seismic data. SEG Technical Program Expanded Abstracts 1996, 1996.

MARFURT, K., KIRLIN, R., FARMER, S. and BAHORICH, M. 3-D seismic attributes using a semblance- based coherency algorithm. Geophysics, vol. 63, no. 4, pp. 1150-1165, 1998.

OPENDTECT, "Netherland Offshore F3 Block Complete," https://www.opendtect.org/osr/Main/NetherlandsOffshoreF3BlockComplete4GB.

RANDEN, T., PEDERSEN, S. and SØNNELAND, L. Automatic extraction of fault surfaces from three- dimensional seismic data. SEG Technical Program Expanded Abstracts 2001, 2001.

RAWAT, W. and WANG, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation, vol. 29, no. 9, pp. 2352-2449, 2017.

RIJKS, E. and JAUFFRED, J. Attribute extraction: An important application in any detailed 3-D interpretation study. The Leading Edge, vol. 10, no. 9, pp. 11-19, 1991.

ROBERTS, A. Curvature attributes and their application to 3D interpreted horizons. First Break, vol. 19, no. 2, pp. 85-100, 2001.

SIMONYAN, K., ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. Proc. International Conference on Learning Representations, 2014.

TINGDAHL. K. and DE ROOIJ, M. Semi-automatic detection of faults in 3D seismic data. Geophysical Prospecting, vol. 53, no. 4, pp. 533-542, 2005.

VAN BEMMEL, P. and PEPPER, R. Seismic signal processing and apparatus for generating a cube of variance values. US Patent 6151555, 2000.

WALDELAND, A. and SOLBERG, A. Salt Classification Using Deep Learning. 79th EAGE Conference and Exhibition, 2017.

WANG, Z. and ALREGIB, G. Fault detection in seismic datasets using hough transform. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.

WANG, Z. and ALREGIB, G. Fault detection in 3D seismic data using the Hough transform and tracking vectors. IEEE Transactions on Computational Imaging, 3(1), pp.99-109, 2017.

WANG, Z., TEMEL, D. and ALREGIB, G. Fault detection using color blending and color transformations. 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2014.

ZHANG, B., LIU, Y., PELISSIER, M. and HEMSTRA, N. Semiautomated fault interpretation based on seismic attributes. Interpretation, vol. 2, no. 1, pp. SA11-SA19, 2014.

ZHIQIANG, W. and JUN, L. A review of object detection based on convolutional neural network. 2017 36th Chinese Control Conference (CCC), 2017.