Fault Detection using Residual Neural Networks

Paulina Wozniakowska[*], Marcelo Guarido[†], Daniel Trad[†], David Emery[†], David W. Eaton[*]

ABSTRACT

This chapter presents an example of automated fault detection using Residual Neural Network. In this work, synthetic data set from the FORCE: Seismic fault Mapping competition was used. We present how the popular image segmentation technique can be used to identify faults on the 2D images exported from the 3D synthetic seismic cube.

INTRODUCTION

Fault identification is one of the main parts of structural interpretation of unconventional and conventional reservoirs. Areas in the vicinity of faults are the common target of conventional hydrocarbon development as one of the common types of petroleum traps. For the unconventional development, pre-existing faults can pose a serious threat as they can potentially be reactivated during hydraulic fracturing operations and result in high-magnitude seismicity (Lei et al., 2019; Mahani et al., 2017). Potential of automatic fault detection using machine learning has been extensively studied over the recent years. Studies confirm the applicability of machine learning algorithms to detect fault structures on the synthetic and real seismic images, even for limited number training of samples (Li et al., 2019). Some studies suggest that Convolutional Neural Networks can be successfully used to obtain more detailed information, including the probability of the predicted faults and their exact orientation (Wu et al., 2019). Other examples have investigated using unprocessed seismic traces (Zhang et al., 2014; Dahlke et al., 2016).

CONVOLUTIONAL NEURAL NETWORKS VS RESNETS

Convolutional Neural Networks (CNNs) are among the most widely used deep learning algorithms for image processing and computer vision (Krizhevsky et al., 2012). Specifically, they are common choice for the image segmentation problem, which is based on the predictions at the pixel level (Gupta et al., 2014).

Main building blocks of the CNNs include: input layer, convolutional layer, pooling layer, fully connected layer and output layer. Input layer corresponds to the vector of input features fed into the algorithm. Convolutional layer is the most important part of every CNN. As its name implies, it is responsible for convolution of the input from the previous layer and sending the filtered information to deeper layers of the network. Pooling layers, perform the pooling operation, which corresponds to the selection of the characteristics from a bigger region in the previous layer. Most common pooling method include averaging or maximum value selection. Fully connected layers are the type of layers in which all neurons from previous layers are connected to every activation unit of the next layer and

[*] Microseismic Industry Consortium, Department of Geoscience, University of Calgary
[†] CREWES, Department of Geoscience, University of Calgary

perform classification based on the features from the previous layer. The output layer of the CNN produces the final result of all the operations performed in the hidden layers (Géron, 2019).

Residual Neural Network (ResNet) is a type of CNN architecture, which, apart from the standard building blocks described above, include additional residual blocks and shortcut connections, which allows for constructing multi-layer networks by reducing the problem of vanishing or exploding gradients that typically affects deep CNNs (He et al., 2015).

FAULT DETECTION AS AN IMAGE SEGMENTATION PROBLEM

In the recently published literature, the binary image segmentation approach was used to detect structural lineaments. Specifically, the information about the location of faults in space is expressed by the output mask layer, in which each pixel is located in a 3D space and classified as ’fault’ or ’not-fault’ accordingly (Haralick and Shapiro, 1985). An example of a seismic cube and its fault-mask cube used to extract pixel labels is shown in the Figure 1.

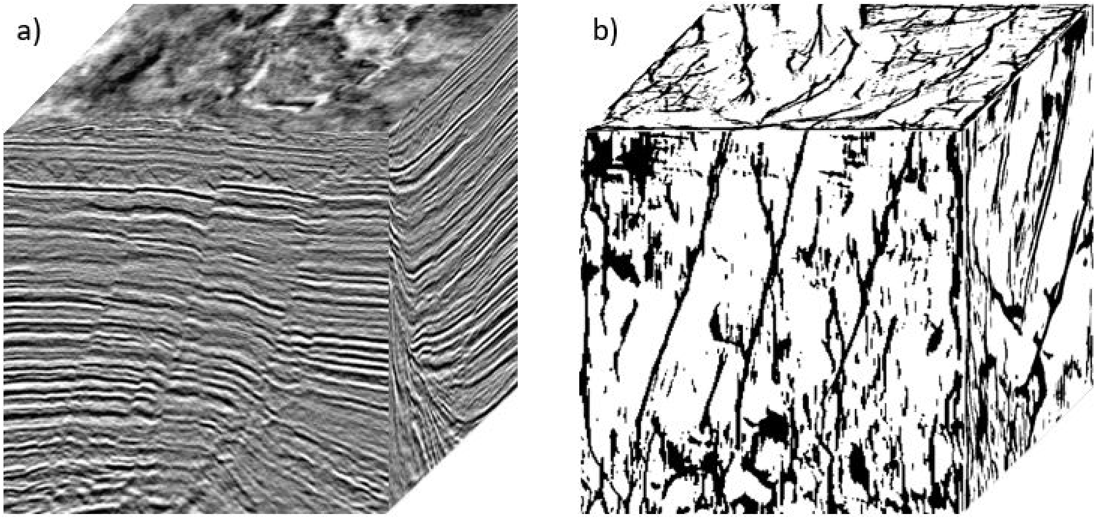
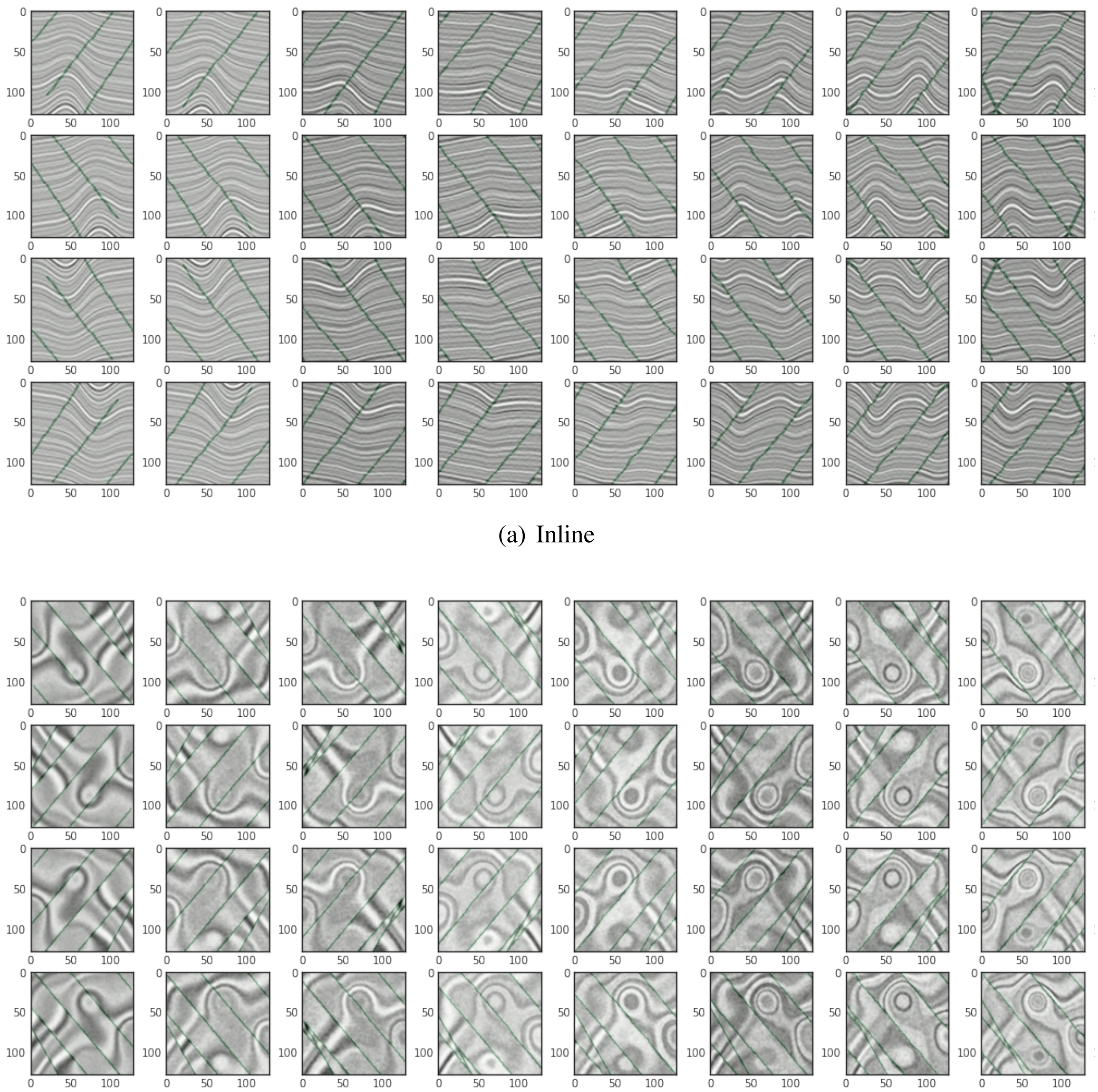


FIG. 1. Example representation of the 3D seimic cube and fault mask (Xiong et al., 2018).

DATA PREPROCESSING

The data used in this study were sourced from the 2020 FORCE: Seismic Fault Mapping competition dataset. It included the synthetic 3D seismic cube of an approximate size of 3km x 3km x 3km. Original 3D Seismic data were preprocessed to obtain the input features and output labels using several steps:

1. **2D image extraction from 3D seismic.** The 3D seismic cube was used to create three separate subsets corresponding to inline and crossline sections as well as horizontal time slices. Each input image had a size of 128x128 pixels. Due to the similarity of the faults represented in inline and crossline directions, inline and crossline



(a) Inline

(b) Time slice

FIG. 2. Examples of image augmentation used in this study. Top rows correspond to 8 inline (a) and time slice (b) sample images exported from 3D seismic and converted to grayscale. Bottom rows present the modified version (horizontally flipped, vertically flipped, and rotated by 180°). 2D seismic images are presented in the greyscale, while green lines represent synthetic faults.

subsets were used to create a model for changes along the horizontal direction (further referred to as inline/crossline model), which resulted in 2-fold increase in the number of the examples available for the training.

2. **Data augmentation**. Number of examples available for training was further increased by implementing the vertical and horizontal flipping as well as rotation of original image by 180° (Figure 2). Augmented dataset consisted of 10400 training samples for each of the inline, crossline and time slice image subsets (Figure 2

3. **Inputs preparation**. Input images were converted into the grayscale and transformed into matrices to generate the input feature vectors. Standard pixel values

in the greyscale are represented by numbers ranging between 0 and 255. Features were therefore normalized to obtain values within the 0-1 range.

4. **Labels preparation**. Labels were generated using the fault masks contained within each of the input image. Pixels were classified as 'fault' or 'non-fault' and assigned 1 and 0 values, respectively, according to the region they represented.

MODEL DEVELOPMENT

Our method was compiled using Tensorflow and Keras - open-source frameworks for machine-learning. It was trained using the binary cross-entropy loss, with RMSprop optimizer. An example of the model architecture is shown on Figure 3

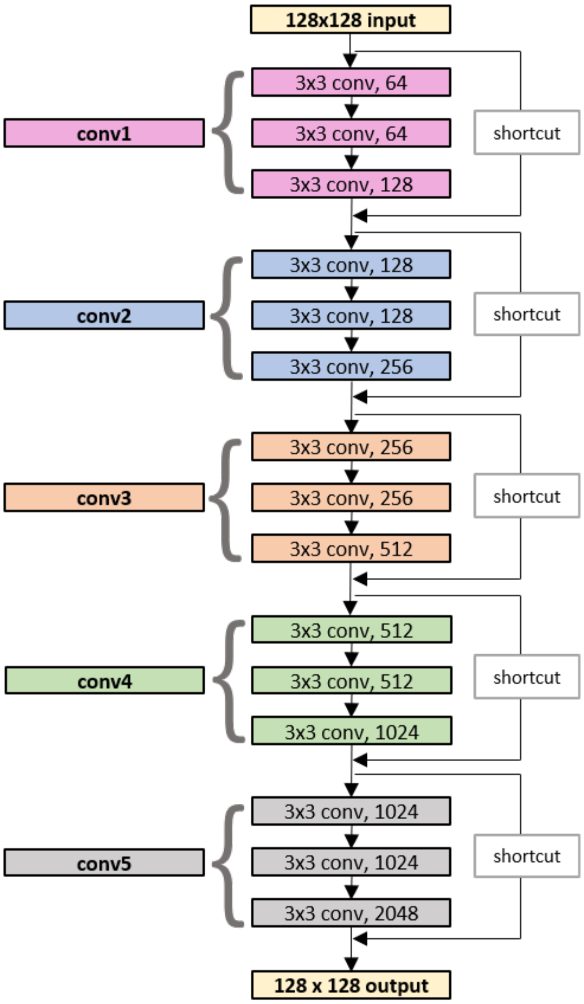
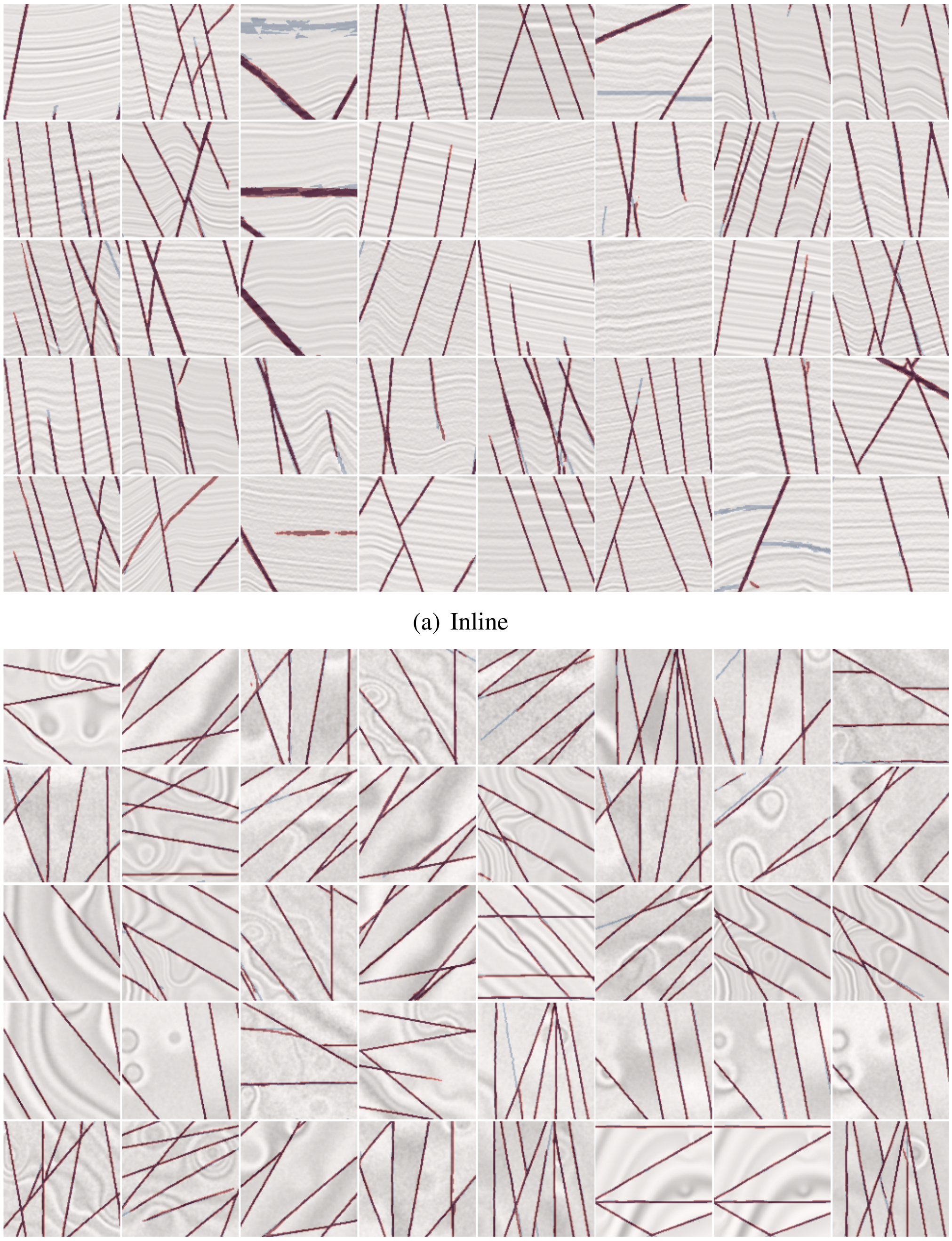


FIG. 3. ResNet architecture used in this study. *3x3 conv* corresponds here to dimension of the filter used. Value after comma indicates the number of channels in the layer. Last layer of each *conv* part has the number of channels increased by 2.

The training was performed on the Google Colab using the GPU cloud service. The training took 2.57 hours for the inline/crossline model and 1.52 hours for the time slice model. The Inline/crossline model yielded a binary cross-entropy loss of 0.0854. The model trained on time slices performed slightly better and resulted in the loss value of 0.0703.



(a) Inline

(b) Time slice

FIG. 4. Predictions obtained using ResNet models trained on inline and crossline (a) and time slice (b) sample images training sets. True fault locations and their predictions are indicated by the blue lines and red lines, respectively.

Figure 4 shows the predictions over the validation set (still part of the synthetic data). On the top are samples from the inline and crossline directions, and on the bottom samples from the time slices. In both cases the model performed with high precision, marking correctly most of the faults (the time slice model is close to perfect), with some minor misclassification.

CONCLUSIONS AND FUTURE WORK

In this study, a Residual Neural Network was used to perform automated fault detection. This approach allows for the identification of structural lineaments in 2-dimensional spaces represented by the inline, crossline and time slice images exported from a synthetic seismic 3D cube. Preliminary results suggest that the workflow can be successfully applied to identify fault structures using only 2D information, which emerges as an alternative to computationally expensive 3D detection approaches. Another advantage of our approach is that it can be applied to areas where 3D seismic data are not available. Further development of this study will include testing the model trained on a synthetic cube on real seismic data as well as implementing a more complex model to perform the fault detection and visualisation in 3D. In addition, various seismic attributes will be tested as potential parameters which could improve the quality of the predictions, especially in the case of real data characterized by significantly lower signal-noise ratio comparing to synthetic models.

REFERENCES

Dahlke, T., Araya-Polo, M., Zhang, C., Frogner, C., and Poggio, T., 2016, Predicting geological features in 3d seismic data: Advances in Neural Information Processing Systems (NIPS), **29**.

Géron, A., 2019, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems: O'Reilly Media.

Gupta, S., Girshick, R., Arbeláez, P., and Malik, J., 2014, Learning rich features from rgb-d images for object detection and segmentation, *in* European conference on computer vision, Springer, 345–360.

Haralick, R. M., and Shapiro, L. G., 1985, Image segmentation techniques: Computer vision, graphics, and image processing, **29**, No. 1, 100–132.

He, K., Zhang, X., Ren, S., and Sun, J., 2015, Deep residual learning for image recognition, 1512.03385.

Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012, 2012 alexnet: Adv. Neural Inf. Process. Syst., 1–9.

Lei, X., Wang, Z., and Su, J., 2019, The december 2018 ml 5.7 and january 2019 ml 5.3 earthquakes in south sichuan basin induced by shale gas hydraulic fracturing: Seismological Research Letters, **90**, No. 3, 1099–1110.

Li, S., Yang, C., Sun, H., and Zhang, H., 2019, Seismic fault detection using an encoder–decoder convolutional neural network with a small training set: Journal of Geophysics and Engineering, **16**, No. 1, 175–189.

Mahani, A. B., Schultz, R., Kao, H., Walker, D., Johnson, J., and Salas, C., 2017, Fluid injection and seismic activity in the northern Montney play, British Columbia, Canada, with special reference to the 17 August 2015 M w 4.6 induced earthquake: Bulletin of the Seismological Society of America, **107**, No. 2, 542–552.

Wu, X., Shi, Y., Fomel, S., Liang, L., Zhang, Q., and Yusifov, A. Z., 2019, Faultnet3d: predicting fault probabilities, strikes, and dips with a single convolutional neural network: IEEE Transactions on Geoscience and Remote Sensing, **57**, No. 11, 9138–9155.

Xiong, W., Ji, X., Ma, Y., Wang, Y., AlBinHassan, N. M., Ali, M. N., and Luo, Y., 2018, Seismic fault detection with convolutional neural network: Geophysics, **83**, No. 5, O97–O103.

Zhang, C., Frogner, C., Araya-Polo, M., and Hohl, D., 2014, Machine-learning based automated fault detection in seismic traces, *in* 76th EAGE Conference and Exhibition 2014, vol. 2014, European Association of Geoscientists & Engineers, 1–5.