# APPLICATION OF DEEP LEARNING FOR FAULT DETECTION IN SEISMIC DATA

S M Shamsul Hoque
School of Geosciences, University of Louisiana at Lafayette
LA 70504

## ABSTRACT

Seismic fault detection by using Deep Learning can be highly beneficiary for the geologists and geophysicists. In this paper, a fully convolutional neural network model is developed to satisfy this need. The model trains from a 1512 field seismic data/images which contain seismic fault. By using semantic segmentation method, the model gives either a non-fault or fault (0 or 1) class for every pixel in the image. Though the validation result could be improved, yet it gives us idea where the training model needs to stop before overfitting the training data. The scope to improve the model prediction is also discussed.

## INTRODUCTION

Seismic image gives idea about the subsurface conditions. Many features (fold, fault, joints, rock deformation etc.) of the subsurface can be known using seismic acquisition methods. Mainly, sound wave is utilized for seismic imaging which is produced by a “source” and then reflected, refracted and diffracted along its way through different stratigraphic layers of the subsurface. Due to the velocity and density difference of the Earth’s layers, the recorded reflected waves give us idea about different characteristics of the subsurface (Castagna 1993).

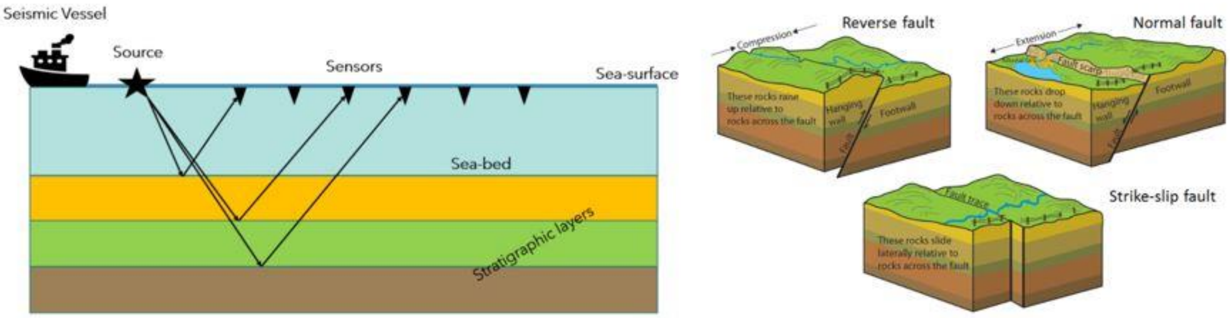


Fig 1. Seismic data acquisition and classification of seismic faults

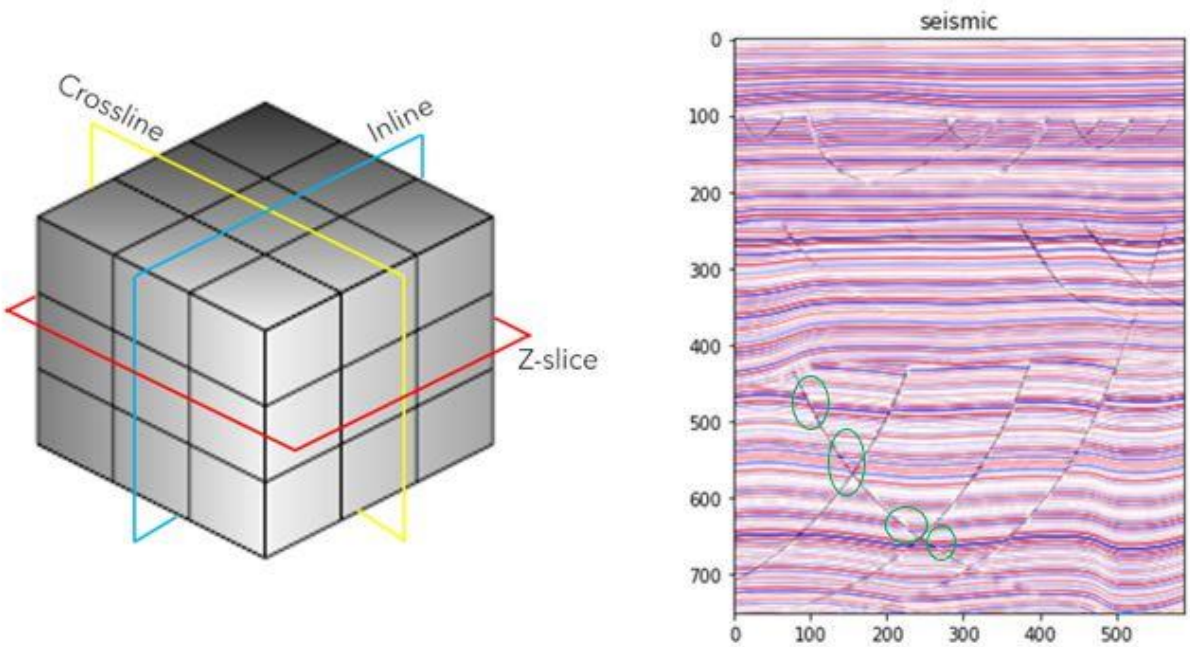


Fig 2. Seismic data volume for analysis

Seismic fault detection is an important step in seismic data processing. Seismic faults are indicator of potential petroleum locations and also they give the geologists evident idea of the previous geologic events in any particular location. It is really a tedious work to locate all the seismic faults of a large seismic dataset manually. Deep Learning can be handy to lessen this huge amount of work (Hinton 2006).

Convolutional Neural Network (CNN) is using successfully at a large for solving different problems of computer vision (Simonyan and Zisserman 2014). Autonomous driving vehicle, medical imaging, object detection etc. are some proved sectors where CNN is useful. Here, in this paper, a U-Net model (a branch of CNN) has been developed for seismic fault detection which follows semantic segmentation process.

In a semantic segmentation procedure, a corresponding label is assigned to every pixel of an image. The neural network model can learn to identify desired characteristics by comparing predicted and actual labeling per pixel. U-Net architecture is used particularly for semantic segmentation task. (Xiong et al, 2018)

For our work, we utilize a large dataset which is originally from a seismic survey called Thebe Gas Field in the Exmouth Plateau of the Carnarvan Basin on the NW shelf of Australia. The dataset contains expert level fault annotations manually done by expert interpreters of faults. Any good model developed by utilizing this large dataset would be very helpful for future seismic fault interpretation process.

.

## DATA PREPARATION

The seismic data and expert annotations are divided into a training set, a validation set, and a test set and stored in three sections- train, validation and test. The whole data contains

1803 crosslines, 3174 inlines with 1537 samples in depth/time axis. The first 900 cross-sections with size of 900 [crossline] × 3174 [inline] × 1537 [sample] are divided into the training set for adjusting algorithm parameters. Similarly, the next 200 cross-sections with the size of 200 [crossline] × 3174 [inline] × 1537 [sample] are divided into the validation set to test the performance of the model during the training process, and the last 703 cross-sections with the size of 703 [crossline] × 3174 [inline] × 1537 [sample] are divided into the test set to evaluate the performance of different algorithms objectively. Due to the large size of the dataset, data are sequentially divided each set into smaller files. Each file contains 100 cross sections. For the time and computational limitation, this work is based on first 600 crosslines from training set and 100 crosslines from validation set.

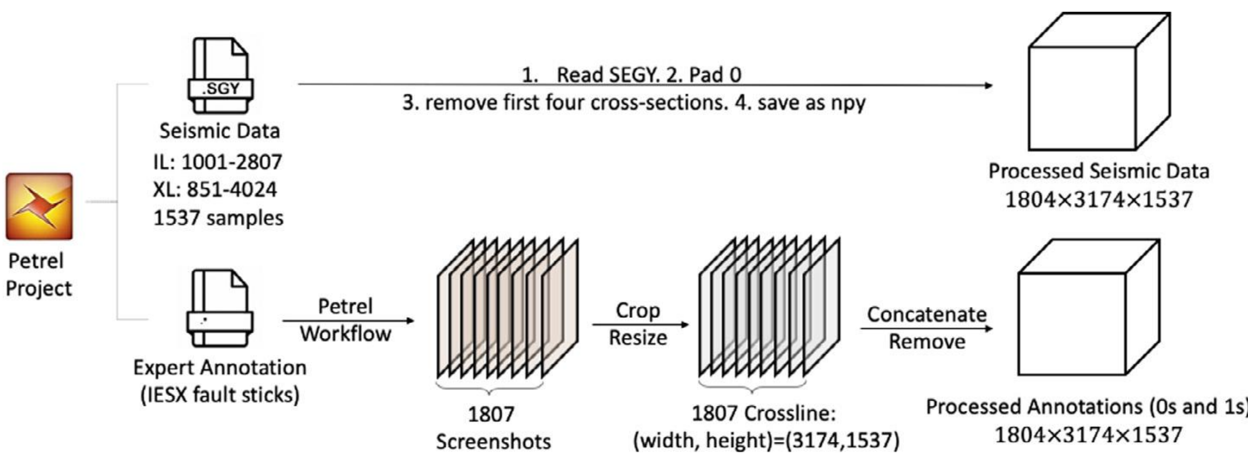


Fig 3. Dataset processing

Though this seismic survey dataset is more than 4 km long in the depth axis, the fault annotations from the interpreters are done from 2 km to 4 km only as this is the zone of interest. So, the training, validation and test data are cut down only from the zone of interest, as the other part don’t contain any data. Also, for the better computational advantage, the depth range is maintained by 512 samples which almost cover the whole depth of that specific zone. On the other hand, in the inline axis, the zone of interest does not cover all those 3174 inlines. For training data, first 2048 inlines cover almost whole zone of interest; whereas, for validation data, inlines from 250 to 2298 cover the zone of interest. The, again, the individual data is cut into 512*512 numpy array format for computational benefit. That means, one individual train and validation data of size 2048*512 is sliced down into four 512*512 2-D arrays.

Some of the processed 512*512 data from both training and validation label set don't contain any fault data, or contains very few fault data. This type of data slow down the learning process as those data can give false positive/false negative values. Some of the 2-D arrays are further removed from processed training where fault annotation percentage is less than 2.5% of the 512*512 image/arrays. In the validation set, fault annotation for every image is greater than 2.5%. After doing all those processing, the shape of the training set and validation set are (1512, 512, 512) and (400, 512, 512) respectively. (The shape denotes that the final training set contains 1512 array/image of 512*512 dimension and the validation set contains 400 array/image of 512*512 dimension).

## METHODS

### Deep Convolutional Network

A deep convolutional network has been used for this work. A generic U-Net framework is shown in figure 2.

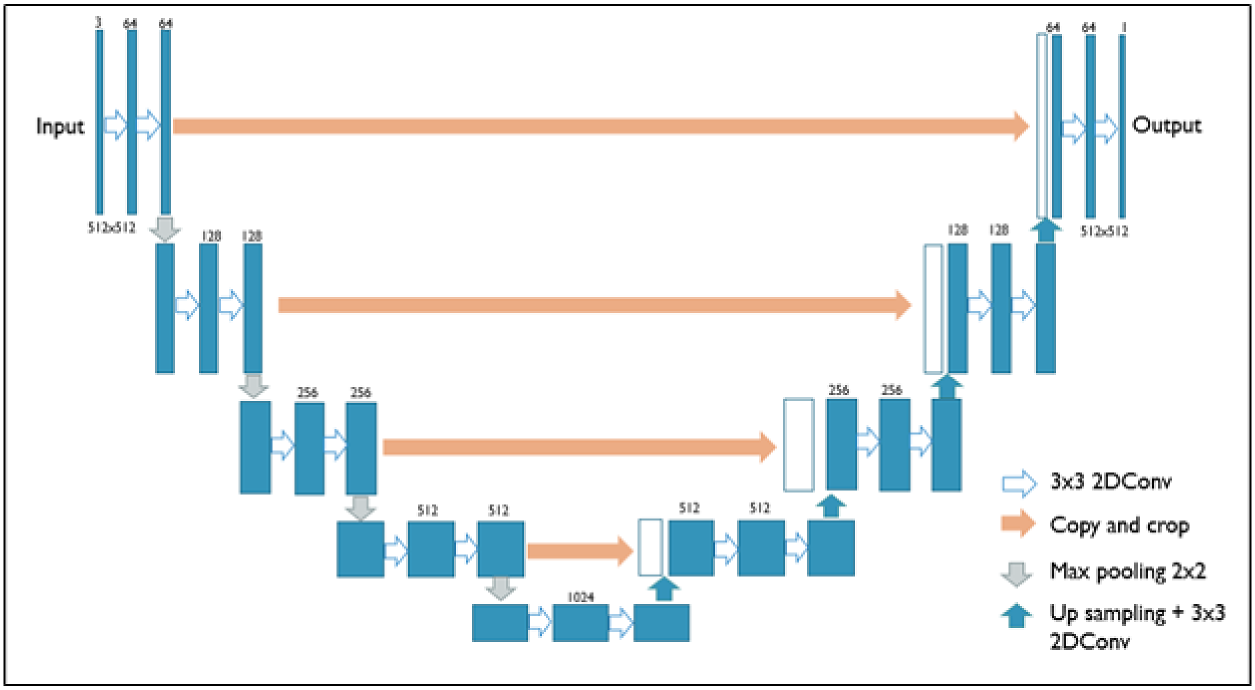


Fig 4. UNet architecture

The forward contraction downsampling process (also known as encoder) involves two 3×3 convolutions followed by a ReLU, a 2×2 max pooling and a stride of 2 for downsampling. As the UNet uses 'unpadded' convolution, the size of the output in every step is in a reduced shape. The reverse upsampling expansion process (also known as decoder) consists upsampling of feature map, a 3×3 convolution and and a concatenation of a feature map from the previous contracting block followed by 3x3 convolutions each with ReLU activation.

**Training and Validation data**

The processed seismic data shape of (1512, 512, 512) is given as input data. The shape means the input data consist of 1512 images/arrays size of 512 × 512. Every image/array can be considered as a batch size because the CNN model for this case takes only one image per iteration. The fault or label data consists either only 0 or 1 (non-fault or fault) for every pixel and this dataset is also supplied to the model during training part.

The same validation data of shape (400, 512, 512) is used repeatedly for validation in every epoch. Comparing training and validation result in every epoch gives idea about the performance of the model and possible required fine tuning of hyperparameters like learning rate, number of epochs etc.

**Model Training**

The model is run on NVIDIA GeForce RTX 3070 with 8 GB memory and with a batch size of 1. The model runs for total 20 epochs. A very slow learning rate of 0.000005 has been used for this CNN model. BCEwithLogitsloss is used as loss function which adds a sigmoid layer and calculate BCE loss in one single pixel. BCE determines the binary cross entropy between the target and the output.

## RESULTS

The validation results give idea when the training model overfits the data. This CNN model validation data loss starts increasing after $8^{th}$ epoch. So, the training model from $8^{th}$ epoch can be used for further test with different dataset.

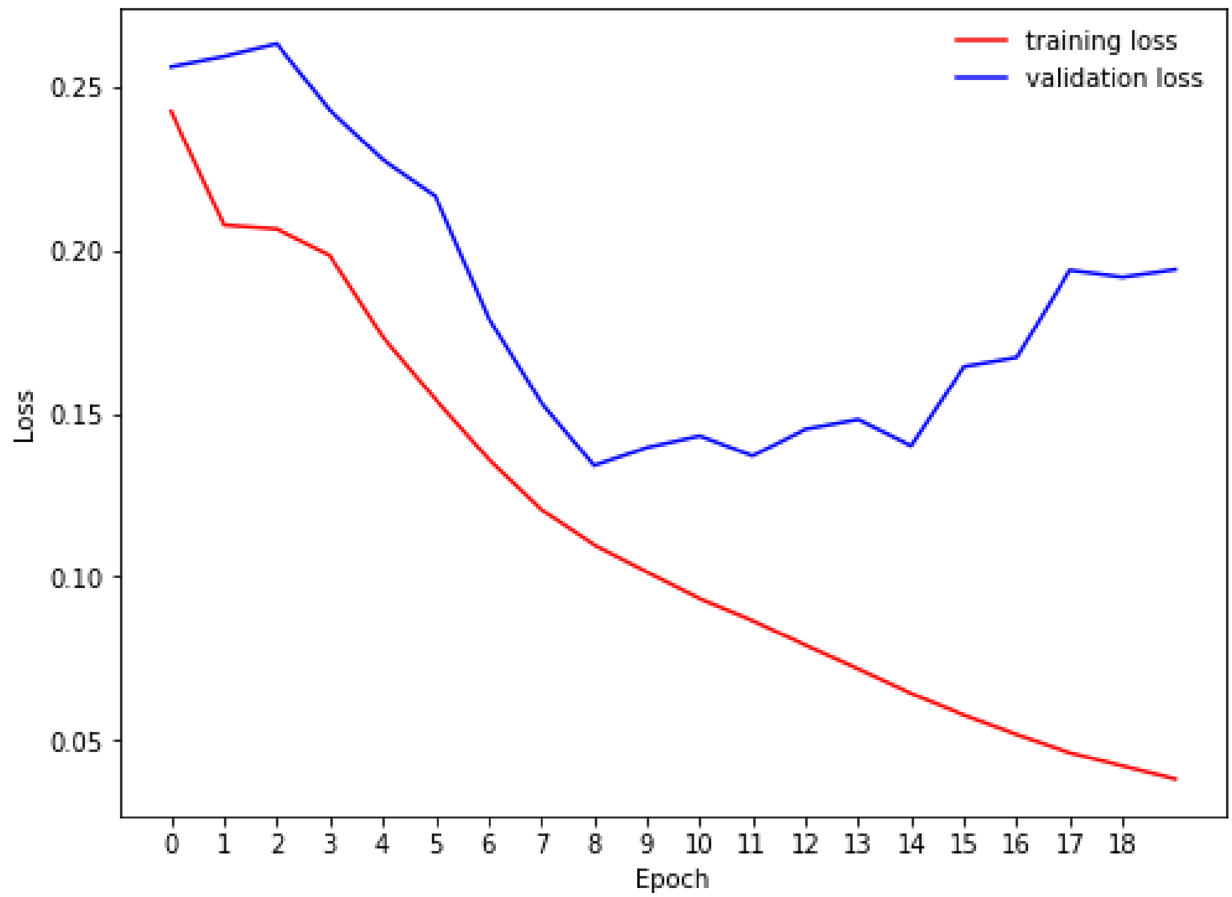


Fig 5. Training and Validation loss plot

The validation loss at 8th epoch shows that there can be still some improvement in the model. Not all the seismic faults are picked correctly by the model.

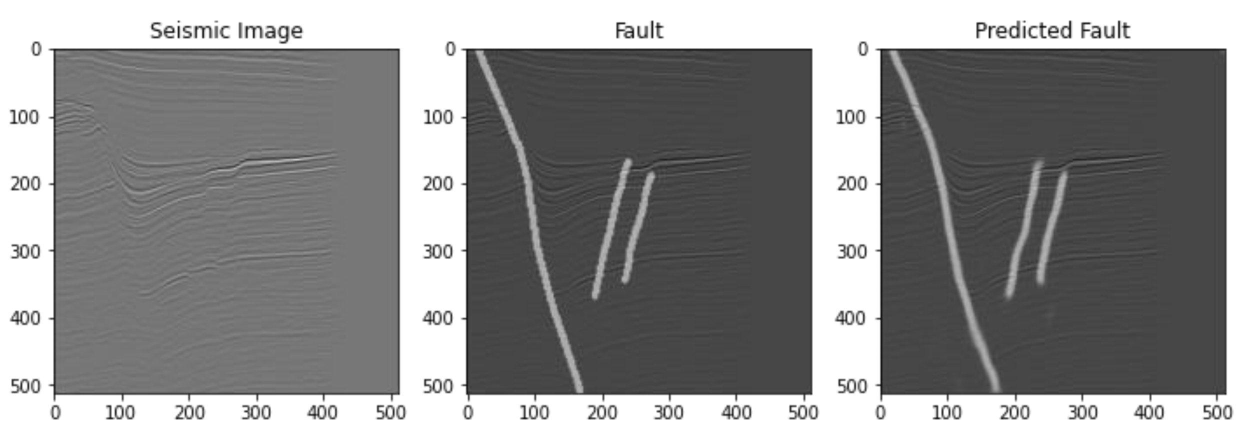


Fig 6. Training result at 19th epoch

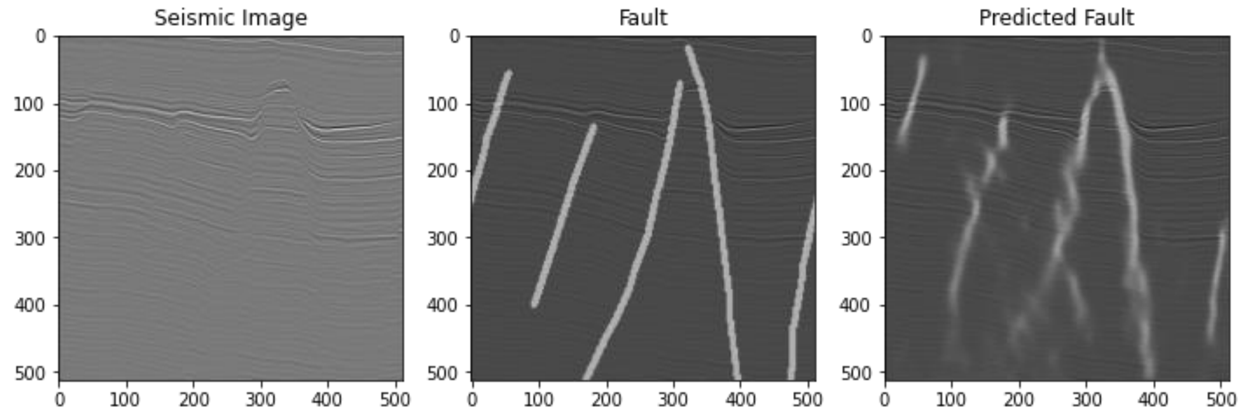


Fig 7. Validation result at $8^{th}$ epoch

## CONCLUSION

The CNN model can pick up the location of seismic faults accurately if it is trained for a larger number of epochs. But the validation result is not satisfactory, which means the model is not generalized for any seismic fault detection dataset. The model can be improved by increasing convolutional layers, adding normalization in the layers or adding dropout layers. For the future direction of works: the model can be improved with necessary modification and then it can be compared with the performance of any transfer learning model like VGG16 or DeepLabv3. The model can also be developed by utilizing all data from this big dataset.

## REFERENCES

Castagna, J. P., 1993, Petrophysical imaging using AVO: The Leading Edge, **12**, 172-179.

Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. science, 2006, 313(5786): 504-507.

Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv, 2014: 1409.1556v6.

Xiong, Wei; Ji, Xu; Ma, Yue; Wang, Yuxiang; BenHassan, Nasher M.; Ali, Mustafa N.; Luo, Yi (2018). Seismic fault detection with convolutional neural network. Geophysics, 1–28. doi:10.1190/geo2017-0666.1