Hydrocarbon Exploration using Seismic Images

Abhishek Varma, Aman Gupta, Dylan Saldanha, Prof. Madhushree

Computer Science and Engineering Department, Sapthagiri College of Engineering (VTU)

Bangalore, India

abhishekvarma16@gmail.com

aman.iv0012@gmail.com

dylxn@protonmail.com

madhushree@sapthagiri.edu.in

***Abstract*— Seismic-data interpretation has as it's main goal the identification of compartments, faults, fault sealing, and trapping mechanism that hold hydrocarbons; it additionally tries to understand the depositional history of the environment to describe the relationship between seismic data and a priori geological information. Finding hydrocarbons through salt detection has been an intricate process, ever since they images salt the first time. Salt bodies form oil traps, which form potential reserves for hydrocarbons. This forms the basis for the motivation behind hydrocarbon sensing via salt detection. Seismic data interpreters are used to interpreting on 2D or 3D images that have been heavily processed. In our problem statement we our dealing with data that is less noisy which is an added advantage. Our solution to the problem is to basically use U-Nets which have shown state of the art results on image segmentation. Each pixel in the image is checked for the presence of salt and further on this is how the proportion of salt in that seismic image is calculated. The energy function is computed by a pixel-wise soft-max over the final feature map combined with the binary cross entropy loss function. Our model boasts an accuracy of 94.7% which is significantly higher than that of in the previous implementations.**

Keywords—U-Nets, CNN, hydrocarbon exploration, salt body identification.

I. INTRODUCTION

Acquisition technology advances and exploration of complex areas are pushing the amount of data to be analysed into the "big data" category. Current exploration workflows consist of many partially automated steps in which domain experts (geologists, geophysicists, rock physicists, etc.) command highly tuned applications and then curate the resulting data in search of valuable information. The data explosion is stressing these workflows to a point at which every year more of the data remains unused. The exploration process can be split up into two elements i.e. advanced tools and manpower. Tools have progressed, and the addition of high-performance computing has helped to reduce turnaround times for seismic imaging [1]. Even in the extreme case in which execution time for processing tools would take nearly zero time, the problem of manpower remains; there is no sensible way in which domain experts can analyse and interpret all incoming data. The best solution must trade domain-expert time for computing time. Therefore, some of that domain knowledge needs to be formalized and implemented within existing and future tools. One way to achieve this is by taking advantage of algorithms that learn, for instance, from legacy data that have been properly vetted. Using machine learning, we can take advantage of new algorithms and software ecosystems, as well as specialized hardware. In this contribution, we will focus our attention on one such application of machine learning.

Seismic imaging is the primary tool used to build high-resolution models of the subsurface. In practice, it is typically part of an iterative workflow that alternates between imaging steps and model update steps. The above refinement is expensive when taling about human costs and computational costs. Most prior work focused on identifying features in already migrated image [2] [3]. The literature is filled with refinements to this workflow, but ultimately, it remains largely the same.

II. BACKGROUND WORKS

A. Supervised learning to detect salt body

In order to strengthen the data interpretation data mining strategies (Hastie, 2011) are employed to classify points or parts of the 3D seismic data. Multiple studies have shown the benefits of using data mining techniques for seismic-data interpretation. Global optimization methods (Shi et. al., 2000; Hale et. al., 2003) are best used to segment a seismic image into structural and stratigraphic geologic units (Hale, 2002). The application of Self Organizing Maps (Castro de Matos et. al., 2007) is another solution that uses unsupervised learning techniques. The authors' new approach is essentially a novel salt body detection workflow. The goal is to create a software solution that can automatically identify, classify and delineate salt bodies from seismic data using seismic attributes and supervised learning algorithms.

In [3], their data analysis phase receives as input a body of seismic data with the task of automatically identifying salt regions. To achieve a class-balanced problem, the authors made sure exactly one half of the subset corresponded to salt, and the other half as non-salt (the task exhibited equal class priors). The authors' model was built using 2 million training voxels. Accuracy is estimated using 10-fold cross validation (Hastie, 2011). This classification model was further used to generate labels for the entire body of seismic data (376,752,501 voxels).

Top performing learning algorithms were the following: Gradient Boosting Trees (Accuracy 80%), Extremely Randomized Trees (Accuracy 80%), and Random Forests (Accuracy 79%). All of their learning algorithms are ensemble methods; these techniques have shown remarkable performance due to their facility to procure low bias (using complex decision boundaries), and low variance (achieved by averaging over various models).

The authors' final predictive model of choice was Extremely Randomized Trees, which was used to predict the labels of 376,752,501 samples; this resulted in a Boolean mask. The accuracy reported was essentially the same as in cross validation (80%). After that, they have removed outliers and misclassification using mathematical morphological operations and a 3D interactive guided (manual intervention) tool developed in house; finally, they used threshold segmentation using local average threshold to get better detection results.

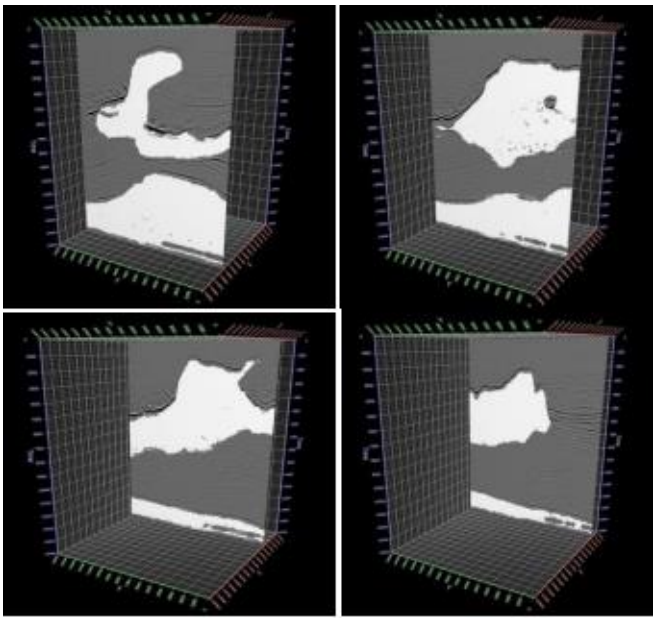

FIGURE 1: Overlapping between seismic data and salt body detected

To measure accuracy, the authors' count the number of matches between the detected salt body and the interpretation by using both volumes, they have counted the number of hits voxel by voxel. The authors refer to this number as NH. The effectiveness ratio is calculated as: (NH/TS) * 100, where TS is the total number of voxels in the volume. Following this technique, they have obtained an accuracy of 95.22%.

B. Automated Fault Detection Without Seismic Processing

The first step in Reference 5's workflow (Figure 4) is to collect the training examples. Instead, the idea here is to generate realistic 3D velocity models synthetically, with the fault labeling generated concurrently for an unbiased ground truth. It is believed by the authors that the randomized models produced in this manner are realistic enough to demonstrate the efficacy of neural network predictions (Figure 4)

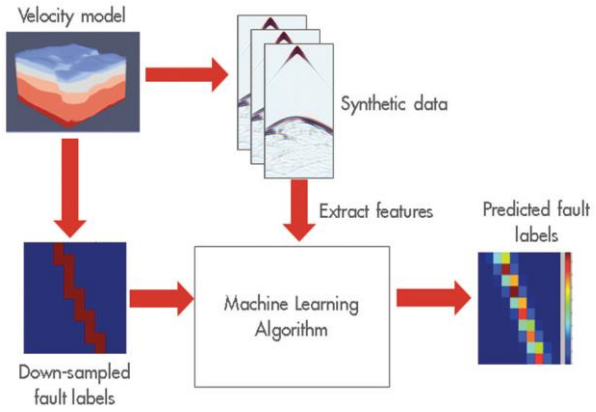

FIGURE 3:. Depiction of the workflow's main tasks

The authors have generated thousands of random velocity models with up to four faults in them, of varying strike, dip angle, and position. Their models had between three and six layers each, with velocities varying from 2000 to 4000 [m/s], with layer velocity increasing with depth. These models were 140 × 180 × 180 grid points at the sampling used for wave propagation (using the acquisition geometry described earlier) but were subsampled to 20 × 20 × 20 and 32 × 32 × 32 for labeling purposes. The raw data collected was reduced aggressively to a feature set that can fit in an NVIDIA K80 GPGPU memory.

With the generated features and labels, a variety of fully connected deep neural networks are trained. The network architecture main parameters varied from two to 20 hidden layers and 256 to 2048 units per layer. They have used the Wasserstein loss function [4] for training. The output of the networks is a subsampled 3D voxel grid, with each voxel's value indicating the likelihood of a fault being present within the voxel. Each of the voxel was binary valued indicating the presence of a fault. The final predictions were generated by taking the likelihood values map output and applying a threshold to it, such that likelihood values above the threshold would be considered having a fault, while those below would not. As a result, a lower threshold can label a lot of voxels as faults, whereas a high threshold labels less.

C. Automatic salt-body classification using deep-convolutional neural network

In this paper an alternative network architecture inspired by Segnet and U-Net, that overcomes the problem by learning to map encoder outputs to final classification labels of image dimension. The architecure used here is composed of a stack of encoder followed by a corresponding decoder stack stack which feeds into a softmax classification layer. Both the encoder and decoder are fully convolutional layers. The salt body labels are generated automatically with the aid of automatic tools.

The network is first trained on selected 2D slices,then the model is validated by predicting salt body location on other unseen slices.

Here the salt body classification is treated as a semantic image segmentation problem with binary classes: the algorithm assigns a salt label to each image pixel based on the shape of the seismic image. While multiple seismic attributes can aid the salt body detection, for simplicity, we only use seismic amplitude as the input in our automated method.The network has the ability to delineate objects

based on their shape despite their small size. Compared to previous network architectures, this encoder-decoder architecture can be trained end-to-end in order to jointly optimize all the model parameters in the network. They report that the key component of their proposed network is the decoder network which consists of a hierarchy of decoders corresponding to each encoder.

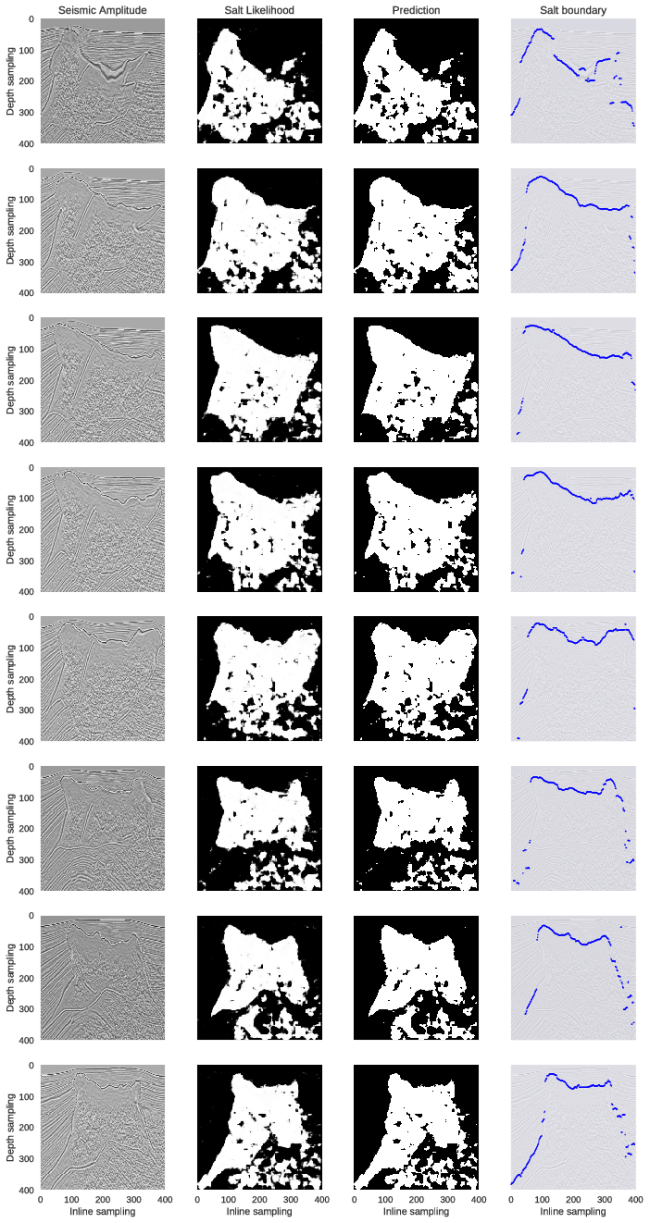

FIGURE 4: Selected crossline test samples and their network output visualizations. The 8 rows represent 8 crossline samples extracted at inline locations different from the ones used in training set: [25, 75, 125, 175, 225, 275, 325, 375]. From left to right, the first column show seismic amplitude images; the second column show probability outputs from softmax classifier of the network; the third column show salt detection prediction results generated by max-likelihood class; the fourth column show top salt boundaries by extracting the first occurence of salt likelihood > 0:6.

III. U-Nets

A. Network Architecture

The architecture contains two paths. First path is the contraction path (also called as the encoder) which is used to capture the context in the image. The encoder is just a traditional stack of convolutional and max pooling layers. The second path is the symmetric expanding path (also called as the decoder) which is used to enable precise localization using transposed convolutions. Thus it is an end-to-end fully convolutional network (FCN), i.e. it only contains Convolutional layers and does not contain any Dense layer because of which it can accept image of any size.

The U-Net architecture [7] is built upon the Fully Convolutional Network and modified in a way that it yields better segmentation in medical imaging. Compared to FCN-8, the two main differences are (1) U-net is symmetric and (2) the skip connections between the downsampling path and the upsampling path apply a concatenation operator instead of a sum. These skip connections intend to provide local information to the global information while upsampling. Because of its symmetry, the network has a large number of feature maps in the upsampling path, which allows to transfer information. By comparison, the basic FCN architecture only had number of classes feature maps in its upsampling path.

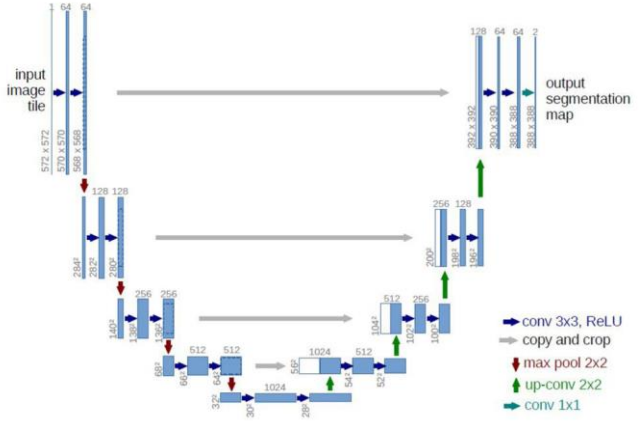

FIGURE 5: U-Net Model

U-Net architecture is separated in 3 parts:

1) The contracting/downsampling path: The contracting path is composed of 4 blocks. Each block is composed of

- 3x3 Convolution Layer + activation function (with batch normalization)
- 3x3 Convolution Layer + activation function (with batch normalization)
- 2x2 Max Pooling

Note that the number of feature maps doubles at each pooling, starting with 64 feature maps for the first block, 128 for the second, and so on. The purpose of this contracting path is to capture the context of the input image in order to be able to do segmentation. This coarse contextual information will then be transfered to the upsampling path by means of skip connections.

2) Bottleneck: This part of the network is between the contracting and expanding paths. The bottleneck is built from simply 2 convolutional layers (with batch normalization), with dropout.

3) The expanding/upsampling path: The expanding path is also composed of 4 blocks. Each of these blocks is composed of

- Deconvolution layer with stride 2

•Concatenation with the corresponding cropped feature map from the contracting path
•3x3 Convolution layer + activation function (with batch normalization)
•3x3 Convolution layer + activation function (with batch normalization)

B. *Network Input*

The raw image is sent to the image preprocessor module and converted from (X, X) into a (101, 101). Further it is upscaled into a (128,128) pixel image. If the image size of size (Y, Y) is lesser than (128, 128) then a black border of 128-Y pixels is added around the image. Further the image is also converted into greyscale.

The pre-processor also converts the image to an array using keras.preprocessing module. It resizes the array to (128,128). Calculates the cumulative mean, cumulative sum of all values across within X(where X is the 2D matrix of the image). It subtracts the cumulative mean from cumulative sum. It divides the cumulative sum by the standard deviation. It finally divides each value in X by 255. The image is converted into an array of pixel values, whose values range from 0 to 1. The array is sent as an input to the trained model.

The user can also choose to give a specific threshold value. If the probability of salt in that pixel is greater than the threshold value then the element's value is left as is. Otherwise, it is replaced with a zero, since the user has previously decided he/she is only interested in regions of salt above a particular probability.

IV. RESULTS

A. *Evaluation Metrics*

In this paper, the proposed model was evaluated using accuracy, rank based statistics, and the mean per class error as stated by Wang et al. [10]. The rank-based evaluations used are the arithmetic rank, geometric rank, and the Wilcoxon signed rank test. The arithmetic rank is the arithmetic mean of the rank of dataset. The geometric rank is the geometric mean of the rank of each dataset. The Wilcoxson signed rank test is used to compare the median rank of the proposed model and the existing state-of-the-art models. The null hypothesis and alternative hypothesis are as follows:

$$H_o : Median_{proposed\ model} = Median_{state-of-the-art\ model}$$

$$H_a : Median_{proposed\ model} \neq Median_{state-of-the-art\ model}$$

Mean Per Class Error (MPCE) is defined as the arithmetic mean of the per class error (PCE),

$$PCE_k = \frac{1 - accuracy}{number\ of\ unique\ classes}$$

$$MPCE = \frac{1}{K} \sum PCE_K .$$

B. *Results*

Figure 6 shows the results obtained by the usage of U-Nets for salt classification. The first column consists of the raw seismic images the model takes as input. The second column shows the mask image which is the ground truth that is used as a comparison for our model's prediction. Both of the above mentioned columns show the X-axis and Y-axis ranging from 0 to 101. This is due to the fact that our dataset consists of images of the same size. The next three columns show the predicted mask image by the U-Net model. We have used three different threshold values 30%, 50% and 90%. It can be observed that these threshold values do make a difference in the prediction. The threshold value is directly proportional the probability of finding salt in the non-black regions.

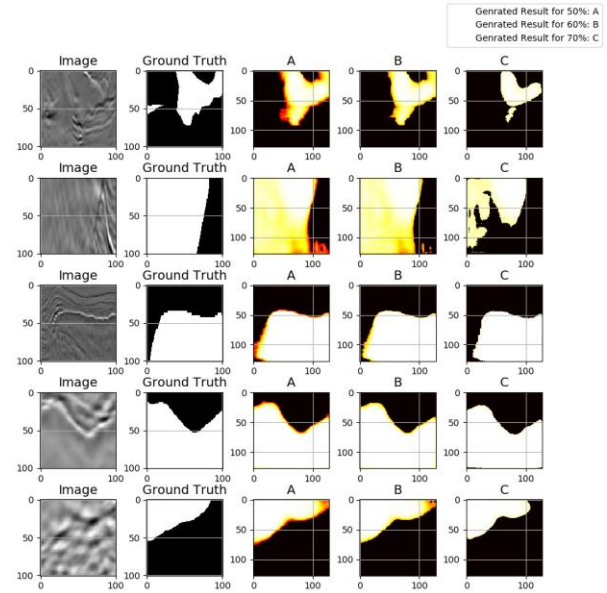

FIGURE 6: Results with three different threshold values- 30, 50 and 90%

V. CONCLUSION

We present a novel approach to the challenging multistep seismic model-building problem. It uses a deep learning system to map out a salt body in the sub surface, using raw seismic recordings as input. A distinguishing aspect of the solution is the use of the U-Net model, which is suited to problems in semantic segmentation. We demonstrate the system's performance on real world data sets with complex salt body formations. The primary challenge going forward will be transitioning to salt bodies with more complex 3D geometry. This will lead to more tuned existing networks and/or extensions to our workflows.

The application of machine learning approaches to seismic imaging and interpretation shows great promise in hydrocarbon exploration and can dramatically change how the vast amount of seismic data is used in the future.

REFERENCES

[1] Rastogi, R., 2011, High performance computing in seismic data processing: Promises and challenges: Presented at HPC Advisory Council Switzerland Workshop 2011, http://www.hpcadvisorycouncil.com/events/2011/switzerland_workshop/pdf/Presentations/Day%203/2_CDAC.pdf, accessed 16 January 2017.

[2] Shi, Yunzhi & wu, Xinming & Fomel, Sergey. (2018). Automatic salt-body classification using deep-convolutional neural network. 10.1190/segam2018-2997304.1.

[3] Guillen, Pablo & Larrazabal, German & Gonzalez, Gladys & Boumber, Dainis & Vilalta, Ricardo. (2015). Supervised learning to detect salt body.

[4] Frogner, C., C. Zhang, H. Mobahi, M. Araya-Polo, and T. A. Poggio, 2015, Learning with a Wasserstein loss: Presented at Advances in Neural Information Processing Systems (NIPS) 28.

[5] Araya-Polo, Mauricio, Taylor Dahlke, Charlie Frogner, Chiyuan Zhang, Tomaso Poggio, and Detlef Hohl. "Automated Fault Detection Without Seismic Processing." The Leading Edge 36, no. 3 (March 2017): 208–214 © 2017 Society of Exploration Geophysicists

[6] Khan, Asifullah. (2016). Introduction to Deep Convolutional Neural Networks.

[7] Ronneberger, Olaf & Fischer, Philipp & Brox, Thomas. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. LNCS. 9351. 234-241. 10.1007/978-3-319-24574-4_28.

[8] Dahlke, T., M. Araya-Polo, C. Zhang, and C. Frogner, 2016, Predicting geological features in 3D Seismic Data: Presented at Advances in Neural Information Processing Systems (NIPS) 29, 3D Deep Learning Workshop.

[9] Lin, H., and M. Tegmark, 2016, Why does deep and cheap learning work so well?: arXiv:1608.08225.

[10] Z. Wang, W. Yan, and T. Oates, ''Time series classification from scratch with deep neural networks: A strong baseline,'' in Proc. Int. Joint Conf. Neural Netw. (IJCNN), May 2017, pp. 1578–1585.