A Survey on Hydrocarbon Exploration using Seismic Images

Abhishek Varma[1], Aman Gupta[2], Dylan Saldanha[3], Madhushree[4]

[1,2,3]*Student, Department of Computer Science and Engg., Sapthagiri College of Engineering, Bangalore, India*
[4]*Assistant Professor, Dept. of Computer Science and Engg., Sapthagiri College of Engineering, Bangalore, India*

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-Net. The energy function is computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function.**

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

1. 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 talking 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.

2. Related work

A. Supervised learning to detect salt body

The primary aim of Seismic-data interpretation is to identify compartments, faults, fault sealing, and trapping mechanism that hold hydrocarbons, it also tries to understand the depositional history of the environment to describe the relationship between seismic data and a priori geological information. Data mining or knowledge discovery in databases (KDD) has become a significant area both in academia and industry. Data mining is the process of extracting novel, useful and understandable patterns from a large collection of data.

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.

1) Automated classification of salt bodies using machine learning

The approach aims at automatically identifying and delineating geological elements from seismic data. Specifically, the authors have focused on the automatic classification of salt

International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

695

bodies using supervised learning techniques. In supervised learning we assume each element of study is represented as an n-component vector-valued random variable $(X1, X2,.., Xn)$, where each Xi represents an attribute or feature; the space of all possible feature vectors is called the input space X. We also consider a set $\{w1, w2,...,wk\}$ corresponding to the possible classes; this forms the output space W. A classifier or learning algorithm typically receives as input a set of training examples from a source domain, $T = \{(xi, wi)\}$, where $x = (x1, x2,...,xn)$ is a vector in the input space, and w is a value in the (discrete) output space.

They assume the training or source sample T consists of independently and identically distributed (i.i.d.) examples obtained according to a fixed but unknown joint probability distribution, $P(x,w)$, in the input-output space. The outcome of the classifier is a hypothesis or function $f(x)$ mapping the input space to the output space, $f: X \rightarrow W$. We commonly choose the hypothesis that minimizes the expected value of a loss function (e.g., zero-one loss).
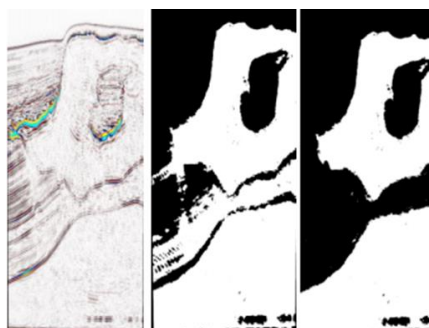

Fig. 1.   (a) Seismic Data (b)Classified Image (c)Results after post-processing step

2) Informative attributes to generate predictive models in seismic data

A proper characterization of voxels can be attained with useful and informative features. We selected three features for our study exhibiting high correlation with the target class: signal amplitude (directly from seismic data), second derivative, and curve length; the last two derived from amplitude. The second derivative is instrumental in the detection of edges, and curve length capture patterns within images, which characterize different features observed inside a salt structure and in its surroundings.

3) Supervised learning algorithms

The authors' data analysis phase receives a body of seismic data as input, with the task of automatically identifying salt regions within the image. To achieve a class-balanced problem, they have made sure exactly fifty percent of the subset corresponded to salt, and the rest 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).

The authors report that Gradient Boosting Trees (Accuracy 80%), Extremely Randomized Trees (Accuracy 80%), and Random Forests (Accuracy 79%) were the top performing learning algorithms. 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).

SEAM I (SEG Advance Modeling Corporation) data, this comes from marine acquisition and represents strong challenges to the geophysical community. The migrated seismic volume was obtained with very low frequency, and there are sediments locations with homogeneous amplitude value than salt body. Our 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 they reported is almost the same as in the case of cross validation i.e. 80%. After that, we have removed outliers and misclassification using mathematical morphological operations and a 3D interactive guided (manual intervention) tool developed in house; finally, we used threshold segmentation using local average threshold to get better detection results.
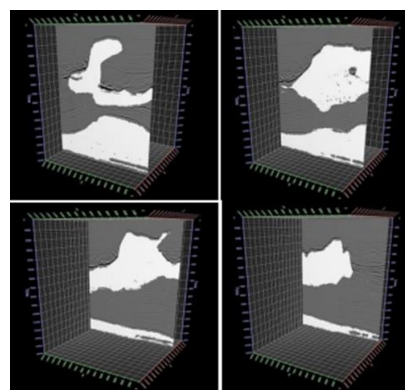

Fig. 2.   Overlapping between seismic data and salt body detected

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. We 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 obtained an accuracy of 95.22%. Machine learning algorithms show highly accurate results, such results are used to predict class labels of voxels on a seismic cube. After the first step, where prediction algorithms are applied directly to the data, the authors had obtained accuracy values of around 80%. They have employed a follow up (post-processing) step that increased the accuracy to around 95%. They finally conclude that when the selected model has high capacity machine learning is a promising mechanism to identify geological bodies on seismic data, and by model averaging it is able to control the variance component of error (using ensemble techniques).
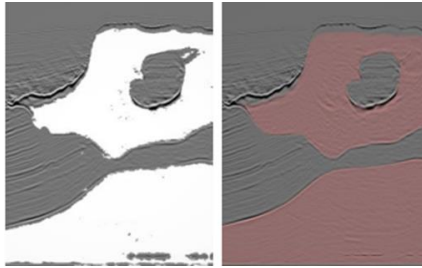
International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

696

Fig. 3. Overlapping between seismic data (a) salt body data (b) interpretation

B. Automated Fault Detection Without Seismic Processing

Recent work [4] demonstrates a new approach that builds and uses a deep neural network (DNN) statistical model to transform raw-input seismic data directly to the final mapping of faults in 2D. The authors here haven chosen fault locations as the output due to their relevance in optimizing production in existing fields. DNNs are built on the premise that they can be used to replicate any function. This paper shows that DNNs can be used to identify fault structure in 3D volumes with reasonable accuracy.

The first step in Reference 5's workflow (Figure 4) is to collect the training examples. Real data examples are impractical and limiting since the labels are assigned to fault locations by few domain experts. This means that the neural network's best result would be bound by human performance and data quality. Instead, the idea here is to generate realistic 3D velocity models synthetically, with the fault labeling generated concurrently for an unbiased ground truth. Next, they use an acoustic approximation to the wave equation to generate wave fields and record them as time-series signals with predefined acquisition geometry. This step is conducted on thousands of random velocity-model realizations, giving many instances of labeled fault locations and the corresponding seismic traces for the entire data set. A portion of this data set is kept unseen from the algorithm (holdout set) so that it can be used for testing after training the predictor. Ranges have been set on a relatively small number of parameters as bounds on the random model generator. These parameters include the number of layers in a model, the number of faults, the range of velocity, and the dip and strike angles for each possible fault. 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).
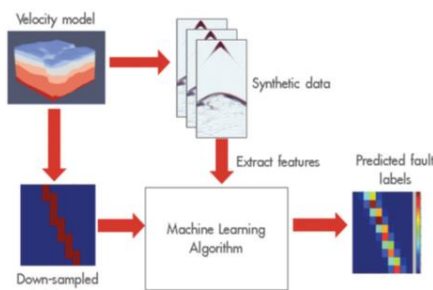
Since the positions of the faults are known a priori, the labelling process is straightforward. However, complexity is added to this process when the labelling targets a subsampled grid for computational efficiency. Labelling of the coarse grid requires an additional threshold parameter. This threshold value sets how many fine-sampled voxels inside a coarse voxel must be fault-labelled for that coarse voxel to be considered as having a fault or not. The threshold is chosen on a trial-and-error basis prior to any training. Raw seismic data is far too unrefined and redundant to be immediately useful as inputs to this neural network. The prediction performance is improved by extracting input features carefully, taking advantage of techniques from signal processing. The amount of collected features is large and grows by orders of magnitude when more realistic models are used.

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 $140 \times 180 \times 180$ grid points at the sampling used for wave propagation (using the acquisition geometry described earlier) but were subsampled to $20 \times 20 \times 20$ and $32 \times 32 \times 32$ for labelling purposes. The raw data collected was reduced aggressively to a feature set capable of fitting 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. For all cases presented in Table 1, they used the Wasserstein loss function 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. Here two different quantitative metrics of performance are used: intersection over union (IoU) and area under the ROC curve (AUC).

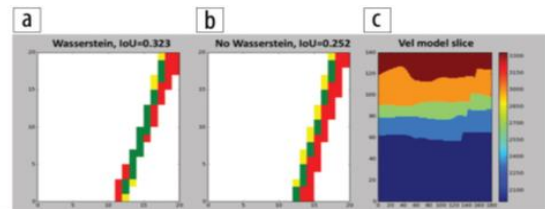
Fig. 4. Depiction of the workflow's main tasks

Fig. 5. Comparison of (a) the Wasserstein- and (b) non-Wasserstein-based predictions, IoU metric (described in the article). Red areas show false positives, green shows true positives (correct predictions), and yellow shows false negative. (c) 2D slice of a 3D model. The predictions have very different IoU, where green means better.

International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
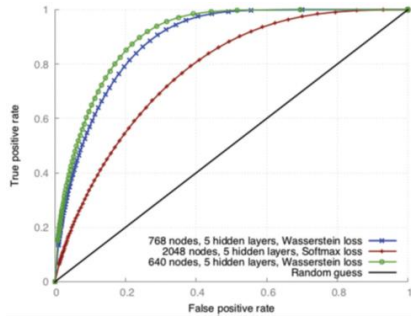**www.ijresm.com | ISSN (Online): 2581-5792**

697

Fig. 6. ROC curves for representative DNN architectures. The closer to the top-left corner the better

The IoU value is a ratio of the number of voxels that are in the intersection of the ground truth and prediction, divided by the number of voxels in the union of the ground truth and prediction. This gives us an idea of how clustered or scattered a prediction is; the values range from 0 to 1, where higher values are better (Figure 5). Two predictions could theoretically have the same AUC value (Table 1) but different IoU values. The (IoU) value averaged for the entire test set of predictions, with predicted likelihoods thresholded at a value chosen to maximize the average IoU over all the predictions.

Each point of a receiver operating characteristics (ROC) curve (Figure 6) is based on the number of true positive predictions (vertical axis) and the false positive predictions (horizontal axis) for a particular threshold value. The AUC for the predictions, which describes how strong our predictor is. The value ranges from 0.5 to 1.0, where the higher the value the better.
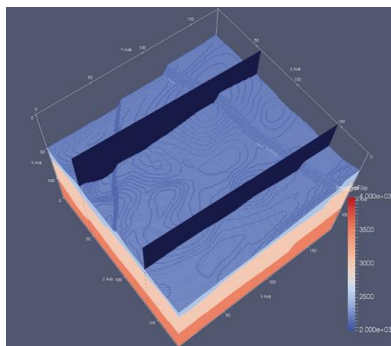

Fig. 7. Example of 3D model with two 2D highlighted slices. In this top view, two faults can be identified

In Table 1, for all data sets (one, two, and four faults in model) AUC exceeds 0.9, which approaches that needed for practical use. Also, IoU surpasses 0.3 for many experiments, which implies that the prediction can be improved in terms of spatial alignment with respect to the ground truth. The prediction grid size of experiments in Fig. 9 is $32 \times 32 \times 32$, therefore each prediction represents a voxel of $4 \times 5 \times 5$ in the model space.

The predictions in Figure 8 follow the expected results, plus some false positives in the bottom-right area of the back slice. However, false positives are present in the area where the faults

coincide. This is in line with expectations, since a cornucopia of signals and patterns are produced in that area. This case exposes the current limit of the author's predicting resolution for complex cases. In general, the resolution is limited by the quality of the data.

Table 1
Results obtained on several representative sets of simulated test data. The first two columns report performance metrics; the other columns describe the parameter of the experiments. All results obtained using Wasserstein loss function, with 16,000 training models and 4000 testing models.

AUC	IoU	Hidden Layers	Nodes per Layer	Faults per Model
0.902	0.311	5	768	4
0.893	0.294	5	640	4
0.836	0.220	7	640	4
0.833	0.218	8	512	4
0.854	0.246	7	512	2
0.849	0.227	6	512	2
0.820	0.219	6	512	2
0.718	0.130	4	1024	1
0.897	0.395	4	512	1
0.919	0.384	4	256	1

To evaluate practical usability, the authors' must address how this approach can be scaled to authentic engenderment-level seismic data sets. Current synthetic input data is predicated on fine-tuned acquisition geometry. If we use this fixed geometry to train a predictor, one must bin and stack the acquired data so that it matches the acquisition used for training the model. For this reason, we believe that using a fixed dense geometry permits us to accurately bin and stack real data to match the predictor's input parameters. Denser acquisition means more features, and as a result, more dramatic reduction of the feature space (and/or more complex neural networks) is needed.
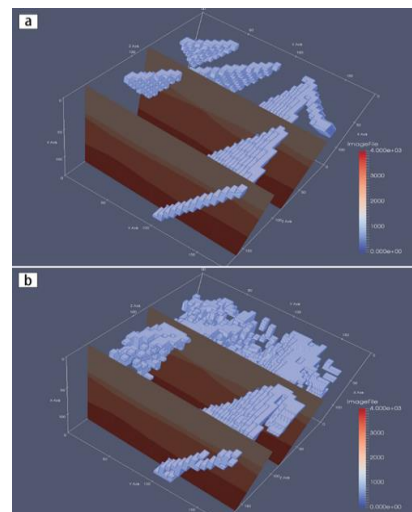

Fig. 8. (a) Expected predictions for fault network in Figure 7 velocity model slices. (b) Author's DDN-based predictions.

The authors report that another area where it can improve the prediction quality is by increasing the count of voxels in down sampled output grid such that we have higher resolution in our predictions. This increase in voxel coarseness means a more computationally demanding neural network and thus an

International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

698

increase in the cost of tuning and training.

This workflow is flexible with respect to what can be predicted. The workflow can be repurposed to build one that predicts, for instance, salt bodies instead of fault locations. It is easy to change the labelling scheme so that salt bodies are labelled before training. Since salt bodies often create strong signals it can be expected that this geologic feature could also be identified using this method. Preliminary results along this line are promising.

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

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 avail the salt body detection, for simplicity, they only utilize seismic amplitude as the input in our automated method. The network has the ability to delineate objects supported their form despite their tiny 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. The key component of our proposed network is the decoder network which consists of a hierarchy of decoders corresponding to each encoder. The original Segnet reuses the max-pooling indices received from the encoders to perform non-linear up sampling of decoder feature maps, but this type of up sampling is prone to checkerboard artifacts (Odena et al., 2016). While these artifacts are acceptable in mundane scale natural images, the issue becomes paramount in seismic images which contain diminutive scale features such as reflection signals, faults and minute discontinuities. To overcome this issue, we adopt the resize convolution proposed by Odena et al. (2016) in our implementation. Each encoder in the encoder network performs convolution with a filter bank to engender a set of feature maps. These are then regularized by batch normalization (Ioffe and Szegedy,2015). Element-wise rectified-linear unit ($\sigma(x) = \max\{0, x\}$) is applied as non-linear activation. Following that, we perform max-pooling and the resulting output is sub-sampled by a factor of 2. Max-pooling achieves translation invariance over minuscule spatial shifts in the input image. The appropriate decoder in the decoder network up samples its input feature maps by resizing-interpolation. This step engenders sparse feature maps which are convolved with a trainable decoder filter bank to engender dense feature maps. Batch normalizations are also applied. The high dimensional feature representation at the output of the final decoder is victualed to a trainable softmax classifier. The output of the classifier is a K channel image of classification probabilities, where K is the number of classes. The k value is set to 2 where pixels are classified as inside/outside salt body.

Training process used here:

The authors' train and test the network utilizing SEAM Phase 1 dataset. This dataset contains a 3D seismic volume which a salt body subsists in the middle. They utilize the 2D single-channel seismic amplitude data as our input to the network. The challenge is to identify the salt body, from a strepitous seismic image. The network can only learn from the subtle features such as high reflectivity and largely dipped boundary. They culled 8 crossline 2D slices as training data. The training labels are manual annotations engendered by optimal path picking method (Wu et al., 2017). Afore each epoch, the training set is shuffled. Compared to the size of the model, this is a fairly minute dataset; however, the bottleneck architecture ascertains that essential relationship are captured. The model weights, or parameters, are initialized utilizing the technique described in He et al. (2015). They utilize the cross-entropy loss as the objective function, and adaptive momentum descent (Adam) as optimization algorithm (Kingma and Ba, 2014) to iteratively update the model weights. After 200 epochs of training, the model achieves 98.77% ecumenical precision (the percentage of pixels correctly relegated in the image). The figure here shows the training results of culled data samples. The salt likelihood is the probability output of the softmax classifier, and the prognostications are assigned by max-likelihood class. Compared to the ground truth, the training precision is proximately as good as the human interpretation.

The input and output shapes are listed in the format: [samples number, image height, image width, channels number]. All samples numbers are not fixed in the network since multiple data samples can run in parallel. Note that the initial input image has 1 channel i.e. the seismic amplitude, and the final output has 2 channels i.e. binary classification. The intermediate channels represent multiple feature maps.

D. Validation tests

The authors' first test the performance of the trained model at different crossline slices. Figure 9 shows the network output of these unseen crossline slices. Since these slices are still crossline slices, they apportion some kindred features with the training data.

It is noticeable that some noisy artifacts appear as "holes" in the detected salt body; however, the global shape of the salt body is extracted accurately, especially on the top boundary of the salt. To visualize the result, they extract the top salt locations with > 0:6 classification probability. The right column in the figure clearly shows these top salt boundaries match seismic amplitudes accurately.

The authors then test the model on inline slices. The inline slices consists of images significantly different from the training set. The performance on this test can imply whether the network successfully learn important features independently to the view perspective. Figure 9 shows the network outputs of these inline slices. The first row of Figure 4 only contains a salt intrusion at the bottom-left corner, but the network seems to falsely classify some of the horizontal reflections as salt. The second to the seventh rows show that the salt bodies are correctly detected; however, the deeper parts of the images where seismic. The 8 rows represent 8 crossline samples extracted at inline locations different from the ones used in

International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

699

training set: [25, 75, 125, 175, 225, 275, 325,375]. From left to right, seismic amplitude images is the first column; probability outputs from softmax classifier of the network is shown in the second column; the third column shows salt detection prediction results generated by max-likelihood class; the fourth column show top salt boundaries by extracting the first occurrence of salt likelihood > 0:6. Images are more noisy are incorrectly assigned to salt body. The eighth row shows that although deeper image part suffers from noise, the salt body in the shallow part is still correctly delineated. Except for the first row example, all examples show good detections of the top boundary of the salt body.
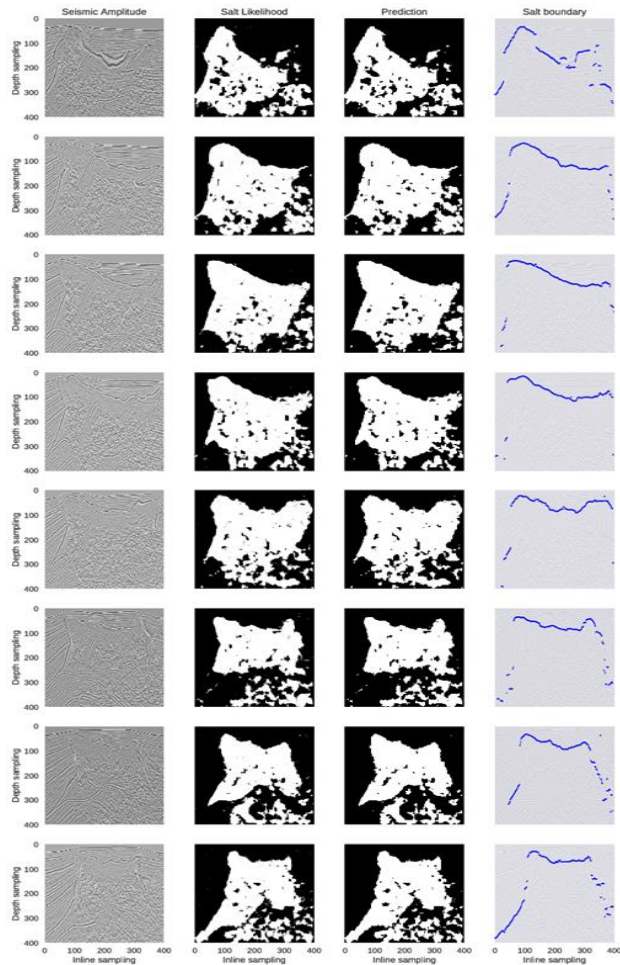

Fig. 9. Selected crossline test samples and their network output visualizations. The 8 rows represent 8 crossline samples extracted at inline locations different from the ones used in training set: [25, 75, 125, 175, 225, 275, 325, 375]. From left to right, the first column show seismic amplitude images; 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

Based on these results, the authors conclude that the proposed method can generalize well even when trained with only small dataset; the top boundary detection is the most robust prediction by the network.
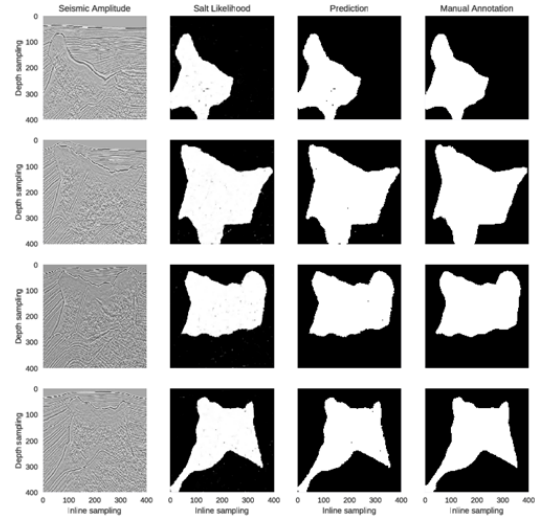

Fig. 10. Selected training samples and their network output visualizations. The 4 rows represent 4 crossline samples extracted at inline locations: [0, 150, 250, 350]. From left to right, the first column show seismic amplitude images; the second column show probability outputs from softmax classifier of the network; the third column show salt detection prediction results generated by max-likelihood class; the fourth column show manual annotations used in the training

3. Conclusion

We present a survey of the different approaches used by various authors in order to solve the challenging multistep seismic model-building problem. Most of them use a deep learning system to map out a salt body in the sub surface, using raw seismic recordings as input.

- In [2], the authors have employed a multi-layered deep convolutional neural network that is capable of capturing subtle salt features automatically without the need for manual input. This model has also been efficiently applied to a whole 3D volume of seismic data.
- In [3], the authors have decided on adopting Extremely Randomized Trees after going through a number of other promising machine learning techniques.
- In [5], A distinguishing aspect of the solution is the use of the Wasserstein loss function, which is suited to problems in which the outputs have spatial layout dependency. The authors demonstrate the system's performance on real world data sets with complex salt body formations.

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

[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.

[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.