Seismic Image Analysis for Salt Detection Using CNN

Dr.Ch Raja[1], Dr. Shaik Irfan Babu[2]

[1]Associate Professor, Dept. of ECE, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India.

[2]Assistant Professor, Dept. of CSEET,Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India.

ABSTRACT

Computer vision is an interdisciplinary field focused on enabling computers to derive high-level understanding from digital images and videos. In recent years, deep learning has significantly accelerated advancements in this domain. One key task in computer vision, known as semantic segmentation, involves labelling each pixel in an image with a corresponding class based on what it represents.In regions rich in oil and gas reserves, large underground salt deposits are commonly found. However, accurately identifying these salt formations remains a challenge. Traditional seismic imaging still relies heavily on expert human interpretation, leading to subjective and highly variable results. More critically, inaccuracies in salt mapping can pose significant risks to oil and gas drillers.This paper addresses this challenge using UNET, an end-to-end fully convolutional network (FCN) designed for semantic segmentation. The seismic image data and corresponding masks are loaded into the system and processed by the UNET model. The network extracts features through successive convolution and pooling layers, gradually down sampling the image. It then reconstructs the original image using transposed convolutions to up sample the feature map

Once trained, the UNET model can accurately analyse new seismic images and delineate areas containing salt deposits, providing a more reliable and automated solution for salt identification in seismic imaging.

Keywords: Fully Convolutional Network (FCN), UNET, Semantic Segmentation, Seismic Imaging

INTRODUCTION

Seismic imaging plays a crucial role in visualizing underground structures and is widely used for discovering hydrocarbon fuel reserves. It operates by emitting sound waves that reflect off subsurface structures, with the reflected signals detected at the surface by receiver devices known as geophones. These signals are then processed to generate a three-dimensional representation of underground rock formations. Seismic images primarily highlight the boundaries between different rock types, as the strength of the reflected signal is proportional to the contrast in physical properties at their interfaces. However, while seismic imaging effectively delineates these boundaries, it provides limited direct information about the internal composition of the rocks themselves.

In hydrocarbon exploration, seismic images help identify potential reservoir rocks, making salt deposit detection particularly important. Salt domes can deform surrounding rock layers, creating natural traps that accumulate oil and gas. The physical characteristics of salt, such as its lower density compared to adjacent rocks, cause strong reflections at its boundaries. While salt deposits often exhibit clear, well-defined edges that are recognizable to the human eye, their identification remains a complex task that relies on expert geologists. Given the massive volume of seismic data, salt-body delineation can take weeks, even with a team of professionals. This challenge makes the problem well-suited for deep learning techniques, which excel at handling large-scale image analysis tasks.

Deep learning has significantly advanced the field of computer vision, particularly in semantic segmentation—an approach that assigns a class label to each pixel in an image. This technology has the potential to revolutionize seismic imaging by automating salt-body identification, reducing subjectivity, and improving efficiency.

This paper aims to address this challenge using UNET, an end-to-end fully convolutional network (FCN) specifically designed for semantic segmentation. The system processes seismic images and corresponding masks through the UNET model, which extracts features via successive convolution and pooling layers, progressively downsampling the data. It then reconstructs the original image using upsampling techniques, such as transposed convolutions. Once trained, the UNET model can analyze any seismic image and accurately delineate areas containing salt deposits, offering a reliable and automated solution for salt identification in seismic imaging.

LITERATURE SURVEY

Traditional methods for salt identification in seismic images rely on teams of expert geologists working for weeks to complete full-survey salt-body delineation due to the vast amount of seismic data. However, this approach presents several challenges:

- **Subjectivity and Variability** – Interpretations can differ among experts, leading to inconsistent results.
- **Limited Availability of Expertise** – Skilled professionals may not always be accessible.
- **Safety Concerns** – Inaccuracies in salt mapping can pose significant risks to oil and gas drillers.

Due to these limitations, seismic image analysis and salt identification have become areas of active research. Traditionally, seismic image analysis followed a hand-crafted approach, where different feature extraction techniques were manually designed and applied to process seismic data.

One of the earliest works in this domain by Pitas and Kotropoulos [1] introduced a method based on texture analysis for semantic segmentation of seismic images, an approach that remains relevant today. Harper and Clapp [2] explored the use of various seismic image attributes to detect salt deposits, while Shafiq et al. proposed a similar method based on new seismic image attribute calculations.

Amin and Deriche [3] developed a technique using a 3D multi-directional edge detector to enhance salt-body identification. Wu [4] introduced a probability-based method to determine salt sediment boundaries, whereas Di et al. [5] proposed a multi-attribute clustering approach using the k-means algorithm for salt delineation.

More recently, Wrona et al. [6] leveraged machine learning techniques to classify four distinct seismic structures by extracting and analysing seismic attributes from images.

These advancements highlight the ongoing shift from traditional, labour-intensive methods toward automated, data-driven approaches for salt identification in seismic imaging.

Design Methodology of Seismic Image Analysis for Salt Detection:

Figure 1 illustrates the architecture of the proposed system, which processes seismic images for salt identification. The system first resizes the input seismic image to 128 × 128 pixels. Next, the pixel values are normalized to a range of 0–1 by dividing by 255, which enhances computational efficiency and preserves image features, enabling the model to make accurate predictions. The preprocessed image is then fed into the UNET model, which has been trained on the TGS dataset. The model performs semantic segmentation, producing a reconstructed image that accurately delineates the areas containing salt deposits.

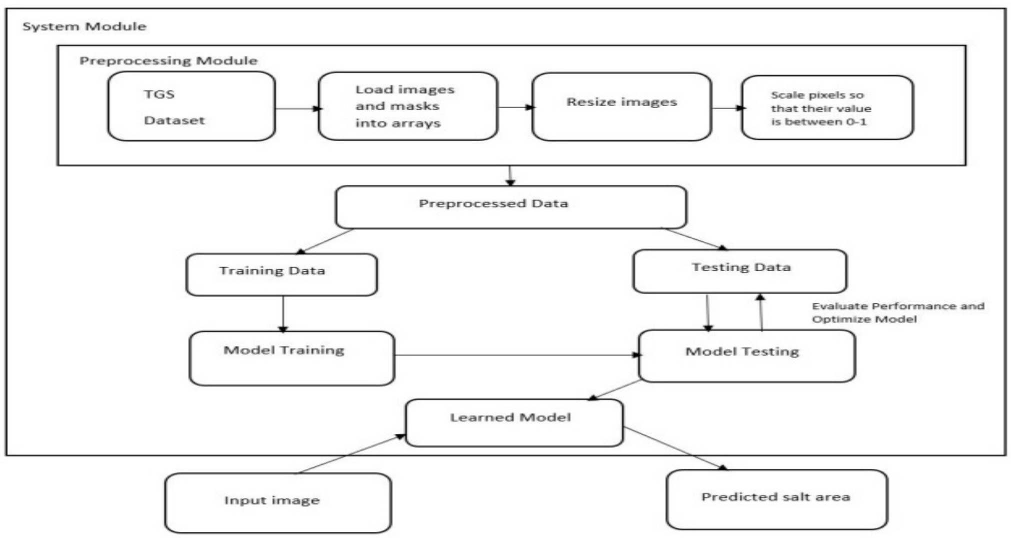

Figure1: System Architecture for Salt Identification in Seismic Images

Figure 2 illustrates the architecture of the UNET model, originally developed by Olaf Ronneberger et al. for biomedical image segmentation. The architecture consists of two main paths:The Contracting Path (Encoder) – This path captures contextual information within the image. It consists of a series of convolutional layers followed by max pooling layers, progressively down sampling the input.The Expanding Path (Decoder) – This symmetric path enables precise localization by using transposed convolutions, gradually up sampling the feature maps to reconstruct the original image dimensions.UNET is an end-to-end fully convolutional network (FCN), meaning it consists solely of convolutional layers without any dense (fully connected) layers. This allows the model to accept images of any size. While the original UNET architecture processes input images of 572×572×3, in this implementation, we use 128×128×3 images. As a result, the feature map sizes at various stages differ from the original paper, but the fundamental components of the architecture remain unchanged.

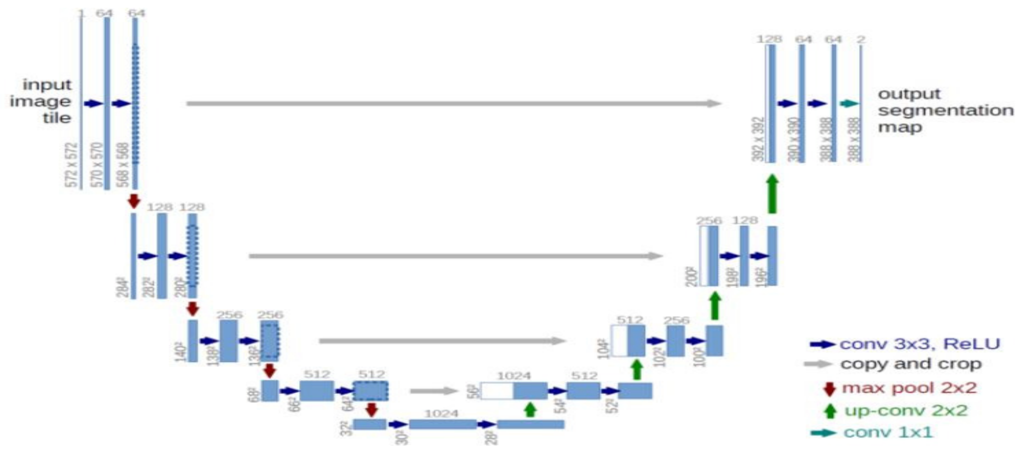

Figure2: UNET architecture

Figure 3 illustrates the modified UNET architecture. In this architecture, 2@Conv layers indicate the application of two consecutive convolutional layers. The notation is as follows:

- c1, c2, ..., c9 – Output tensors of convolutional layers.
- p1, p2, p3, p4 – Output tensors of max pooling layers.
- u6, u7, u8, u9 – Output tensors of upsampling (transposed convolution) layers.

The left side of the architecture represents the contracting path (Encoder), where regular convolutional and max pooling layers are applied. As the image passes through this path, its spatial size gradually decreases while its depth increases, transitioning from 128×128×3 to 8×8×256. This process enables the network to learn "what" is present in the image but results in the loss of "where" information.The right side represents the expanding path (Decoder), which applies transposed convolutions along with regular convolutions. Here, the spatial size progressively increases while the depth decreases, restoring the image from 8×8×256 back to 128×128×1. The decoder is responsible for recovering the "where" information, ensuring precise localization of features.To improve accuracy, skip connections are used at each decoder stage. These connections concatenate feature maps from the encoder at the same level:

- u6 = u6 + c4
- u7 = u7 + c3
- u8 = u8 + c2
- u9 = u9 + c1

After each concatenation, two consecutive convolutional layers are applied to refine the feature maps and enhance precision. This design gives the architecture its characteristic U-shape, leading to the name UNET.

At a high level, the process follows this transformation:
Input (128×128×1) → Encoder (8×8×256) → Decoder → Output (128×128×1).

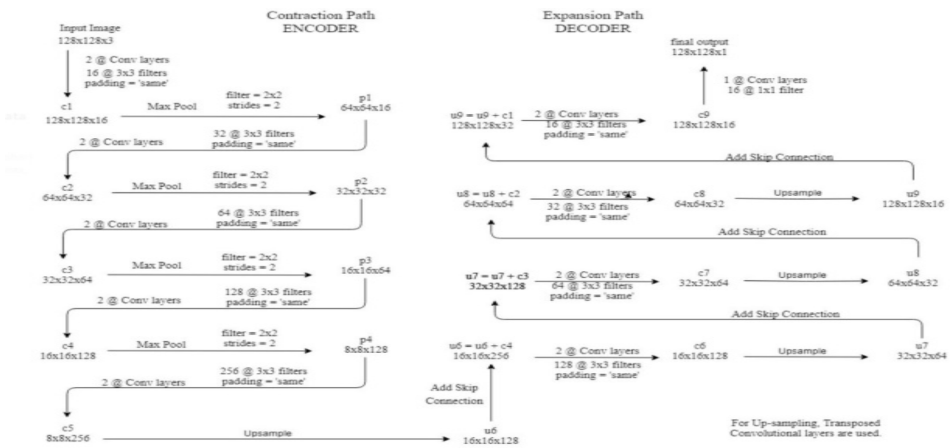

Figure.3: Modified UNET architecture

Convolutional Layer:

In Figure.4, at the core of the UNET architecture is the convolutional layer, which defines the network's functionality. This layer performs a process known as convolution, aimed at extracting high-level features—such as edges and patterns—from the input image.

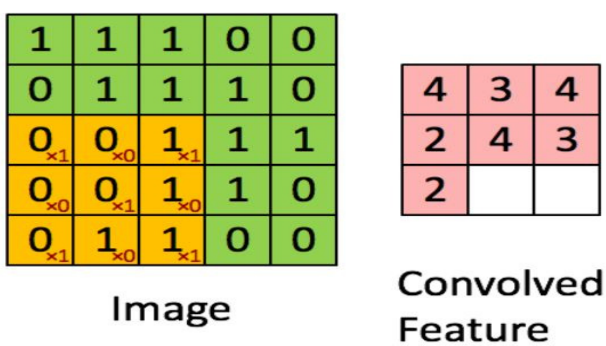

Figure.4: Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

Convolution in UNET

In the UNET architecture, convolution is a fundamental operation that functions similarly to a linear transformation in traditional neural networks. As illustrated in Figure 4, this process involves multiplying a set of weights with the input data. Since UNET is designed for two-dimensional inputs, this multiplication occurs between a 2D array of input data and a 2D array of weights, known as a filter or kernel.The filter is smaller than the input and slides across it, performing an element-wise multiplication between the filter-sized section of the input and the filter itself. The results are then summed to produce a single scalar value, a process referred to as the dot product. Because each application of the filter generates a single value, the operation is sometimes called the scalar product.Using a filter smaller than the input is intentional, as it allows the same set of weights to be applied across different regions of the input. The filter moves systematically from left to right and top to bottom, scanning overlapping sections of the image.This approach is highly effective because a filter designed to detect a specific feature (such as edges or textures) can identify that feature anywhere in the image, rather than at a fixed location. This property is known as translation invariance, meaning the model focuses on whether a feature is present rather than where it appears in the image.Each time the filter is applied to the input, a single value is generated. As the filter moves across the image, it produces a 2D array of output values, forming a feature map that highlights the detected patterns.

Pooling Layer

Like the convolutional layer, the pooling layer plays a crucial role in processing the extracted features. Pooling is responsible for reducing the spatial size of the feature maps by systematically selecting specific values within a window and discarding the rest. This reduces computational complexity and helps extract dominant features that are rotation- and position-invariant, improving the model's ability to generalize effectively.There are two main types of pooling:

1. Max Pooling – Retains only the maximum value within each filter-sized region of the image, as shown in Figure 5.
2. Average Pooling – Computes the average value of all pixels in the filter-sized region.

Both methods help refine the feature maps while reducing their dimensions, making the training process more efficient and effective.

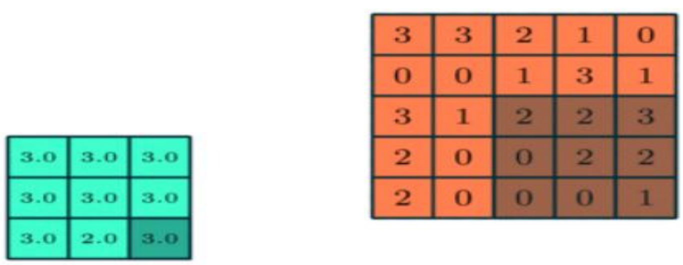

Figure 5:3x3 max pooling over 5x5 convolved feature

1) Max Pooling vs. Average Pooling

Max Pooling not only aids in dimensionality reduction but also acts as a noise suppressor by eliminating noisy activations. This helps in de-noising the data while preserving dominant features. On the other hand, Average Pooling primarily focuses on dimensionality reduction, without explicitly removing noise. Consequently, Max Pooling is often preferred over Average Pooling, as it enhances the model's ability to extract robust features.

2) ReLU (Rectified Linear Unit)

The Rectified Linear Unit (ReLU) is the most widely used activation function in deep learning models. It outputs zero for any negative input, while retaining the same value for positive inputs.ReLU activation functions were originally introduced to differentiate between specific excitation and unspecific inhibition in the neural abstraction pyramid, which was trained to perform multiple computer vision tasks. Unlike traditional sigmoid or tanh activation functions, ReLU allows deep neural networks to be trained efficiently without requiring unsupervised pre-training.Due to its ability to prevent vanishing gradient problems, ReLU enables faster and more effective training of deep neural networks on large and complex datasets.

3) Up sampling with Transposed Convolutions

Unlike traditional image classification, where the output is a class label or bounding box, semantic segmentation generates a full-resolution image, where each pixel is classified into a specific category. However, using a standard convolutional network with pooling layers and dense layers results in the loss of spatial (WHERE) information, retaining only the semantic (WHAT) information.

To recover spatial information, up sampling is required. Several techniques exist for up sampling images, including:

- Bilinear Interpolation
- Cubic Interpolation
- Nearest Neighbor Interpolation
- Unpooling
- Transposed Convolution

Among these, transposed convolution is the preferred method in state-of-the-art neural networks for image segmentation.

4) Transposed Convolution

Transposed convolution follows the same connectivity pattern as standard convolution, but in reverse. This operation allows for up sampling while maintaining learnable weights, eliminating the need for predefined interpolation methods.Despite its name, transposed convolution is not the direct transpose of a convolution operation. Instead, it maps a low-resolution input to a high-resolution output by expanding the input in a structured manner. Unlike traditional convolution (which follows a many-to-one mapping), transposed convolution works in a one-to-many fashion, helping to restore fine details in the image.Some explanations of transposed convolution describe an approach where zeros are inserted between input values before applying a normal convolution to simulate the up sampling effect. However, this method is computationally inefficient compared to directly learning the transposed convolution filters.

Results and Discussion

Testing is a critical phase in software development, ensuring that the system functions as expected and meets the specified requirements. It involves executing the program to identify gaps, errors, or missing functionalities in comparison to expected outcomes. Effective testing helps in debugging, validating, and improving the system's overall reliability.

Table 1: Test Cases

S.No	Filter Size	Pooling Window Size	Dropout	Accuracy on test set
1	2x2	2x2	20%	85.2
2	4x4	2x2	10%	83.6
3	4x4	2x2	5%	84.3
4	3x3	3x3	10%	86.7
5	3x3	3x3	15%	87.3
6	3x3	2x2	5%	92.8

Hyperparameter Optimization and UNET Functionality

Through extensive testing, the optimal hyperparameters for the UNET model were identified, as listed in the last row of Table 1.

The UNET architecture operates in two primary phases:

1. Learning Phase – During training, the model learns to perform semantic segmentation by analyzing patterns in the dataset.
2. Operating Phase – In inference mode, after applying the same pre-processing steps as used during training, the UNET model takes a new seismic image as input and predicts the salt deposit regions.Dataset Selection for Model Training

The dataset used for training the model is provided by TGS, a leading Geoscience and Data company specializing in seismic imaging and 3D subsurface renderings to locate oil and gas reservoirs.This publicly available dataset contains 4,000 seismic image patches, each with a resolution of 101x101 pixels, along with their corresponding segmentation masks (ground truth labels). These masks help train the deep learning model to accurately delineate salt deposits in seismic images.

Understanding the TGS Dataset

Figure 6 illustrates an example from the TGS dataset:

- The left image is a seismic scan. A black boundary has been overlaid for clarity to indicate the salt and non-salt regions (this boundary is not part of the original image).
- The right image represents the segmentation mask, serving as the ground truth label. In this mask:
    - White regions denote salt deposits
    - Black regions represent areas with no salt

The goal of the UNET model is to predict this mask given a seismic image, effectively segmenting salt-rich regions from the subsurface.

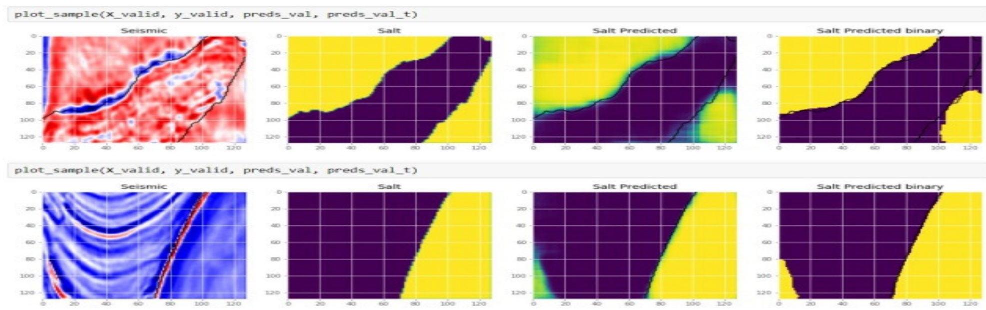

Figure 6: Testing the UNET with new images

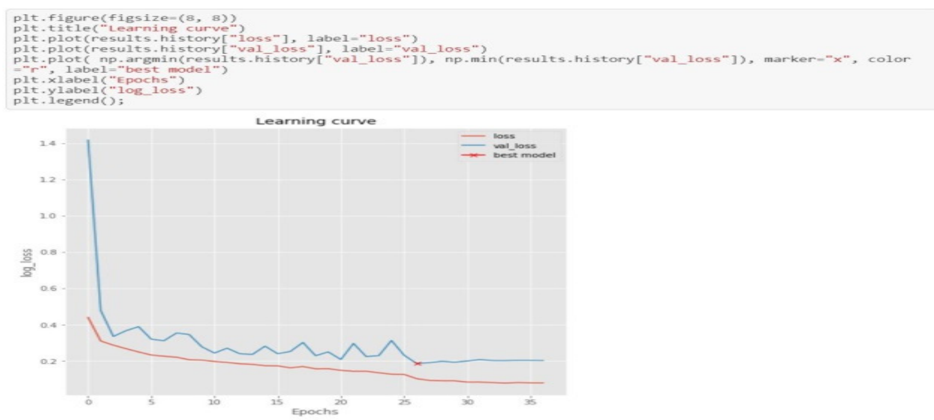

Figure 7: Graph showing model loss

Figure 7, the graph shows how the binary cross entropy or log loss changes after each epoch. We can see how the loss decreases both in the training set and test/validation set. The x marked spot is the best model which has been saved. The results can be tabulated as shown in Table 2.

Table 2: Performance of the Model

Total Number of Samples	4000
Number of Training Samples	3600 (90%)
Number of Testing Samples	400 (10%)
Accuracy on training set	95.5%
Accuracy on testing set	92.8%
UNET error	7.2%

Conclusion

Salt identification from seismic images is a complex task due to the unique characteristics of salt sediments. While their lower density compared to surrounding rocks can create a sharp reflection at the boundary, this very feature makes their identification both simple and challenging. Additionally, the sheer volume of seismic data necessitates weeks of manual analysis by expert geologists to fully delineate salt bodies. This traditional method is subjective, prone to high variability, and, more critically, can lead to potentially hazardous situations for oil and gas drillers due to its inconsistent reliability.To address these challenges, automation of the process is crucial. This paper achieves automation by utilizing UNET, an end-to-end fully convolutional network (FCN) designed for semantic segmentation. The model is trained to analyse seismic images and accurately identify salt deposit regions.

References:

[1] Pitas I., Kotropoulos C. “A texture-based approach to the segmentation of seismic images”. Pattern Recognit.25, 929–945, 1992.

[2] Harper A., Clapp R.G. “Salt Body Segmentation with Dip and Frequency Attributes”. Stanford Exploration Project, Report SEP134, 2008.

[3] Asjad A., Mohamed D. “A new approach for salt dome detection using a 3D multidirectional edge detector”. Appl. Geophys. 12, 334–342, 2015.

[4] Wu X. “Methods to compute salt likelihoods and extract salt boundaries from 3D seismic images”. Geophysics 81, 119–126, 2016.

[5] Di H., Shafiq M., AlRegib G. “Multi-attribute k -means clustering for salt-boundary delineation from three-dimensional seismic data”. Geophys. J. Int., 215, 1999–2007, 2018.

[6] Wrona T., Pan I., Gawthorpe R.L., Fossen H. "Seismic facies analysis using machine learning". Geophysics, 83, 83–95, 2018.

[7] Aleardi, M., Tognarelli, A., and Mazzotti, A., 2016, Characterisation of shallow marine sediments using high-resolution velocity analysis and genetic-algorithm-driven 1D elastic full-waveform inversion, Near Surf. Geophys., 14(5), 449-460, doi: 10.3997/1873-0604.2016030.

[8] Araya-Polo, M., Dahlke, T., Frogner, C., Zhang, C., Poggio, T., and Hohl, D., 2017, Automated fault detection without seismic processing, Lead. Edge, 36(3), 208-214, doi: 10.1190/tle36030208.1.

[9] Bahorich, M., and Farmer, S., 1995, 3-D seismic discontinuity for faults and stratigraphic features: The coherence cube, Lead. Edge, 14(10), 1053-1058, doi: 10.1190/1.1437077.

[10] Blacquiere, G., and Nakayama, S., 2019, Optimum seismic acquisition geometry design with the help of artificial intelligence, 89th Ann. Internat. Mtg., Soc. Expl.Geophys., Expanded Abstracts, 117-121, doi: 10.1190/segam2019-3202558.1.

[11] Chang, D., Yong, X., Yang, W., Wang, Y., and Guo, T. C., 2019, U_Net& Residual Neural Networks for Seismic Fault Interpretation, 81st EAGE Conference and Exhibition incorporating SPE EUROPEC 2019, doi: 10.3997/2214-4609.201901387.

[12] Clay, S., and Vardy, M., 2018, Optimisation of Source and Receiver Geometry for Ultra High Frequency Seismic Data, 80th EAGE Conference and Exhibition Incorporating SPE EUROPEC 2018, doi: 10.3997/2214-4609.201802673.

[13] Cozman, F. G., Cohen, I., and Cirelo, M. C., 2003, Semisupervised learning of mixture models, 20th International Conference on Machine Learning, Expanded Abstract, 99-106.

[14] Di, H., Shafiq, M. A., Wang, Z., and AlRegib, G., 2019, Improving seismic fault detection by super-attribute-based classification, Interpretation, 7(3), SE251-SE267, doi: 10.1190/INT-2018-0188.1.

[15] Dy, J. G., and Brodley, C. E., 2004, Feature selection for unsupervised learning, J. Mach. Learn. Res., 5, 845-889.

[16] Fukushima, K., 1980, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybern., 36(4), 193-202, doi: 10.1007/BF00344251.