# A CNN-Based Framework to Classify Anticlines Structures on Seismic Data

**Pablo Guillen Rondon**
**Hewlett Packard Enterprise Data Science Institute,**
**University of Houston**
**Houston, Texas**

## Abstract

Machine learning techniques have been recognized as a potential tool for supervised classification analyses in different domains of research. In this paper, an attempt has been made to classify 2D synthetic seismic images correspond to anticlines structures for water and gas, respectively. The results established the superiority of convolutional neural networks (CNN), a deep-learning algorithm, compared to other classifiers in terms of classification accuracy. The selection and optimization of associated parameters with the machine learning techniques has also been investigated here. The results indicate that the proposed framework based on CNN is a promising mechanism to automatically detect and identify a high percentage of anticlines structures on seismic data.

## 1. Introduction

The evaluation of structures on the subsurface is an important aspect since all hydrocarbons are contained in some kind of structure. Many types of structures are created by folding and faulting and are called structural traps; among these are the anticlines. The anticlines patterns range from simple to exceedingly complex, and evaluation requires the development of new types of computational approaches to extract the useful and valuable underlying information for interpretation.

Machine and deep-learning algorithms play an important role to train the computer system as an expert, which can be used further for prediction and decision making. Machine learning is the field of study that provides computers the ability to learn without being explicitly programmed [1]. Machine learning applied to seismic data is a rapidly growing research area and provides a commendable technique for creation of classification and automatic decision-making. Artificial intelligence (AI) is the main domain and machine learning and deep learning works under this domain. The AI is the major field to display human intelligence in a machine. Machine learning is used to achieve artificial intelligence, while deep learning is a technique used to implement machine learning [2].

Machine learning and especially deep-learning technologies are powerful for mining features or relationships from data, which makes them quite suitable for learning from human experience [3]. One of the most popular deep-learning technologies is the convolutional neural network (CNN), with successful applications in image recognition and classification [4]. There is also increasing interest in applying machine learning or deep-learning technologies to seismic data processing and interpretation [5]. Such techniques can automate the identification of

compartments, faults, fault sealing, and trapping mechanism that hold hydrocarbons. An assortment of studies exists showing the benefit of machine learning for seismic data interpretation [6-16]. The richness and rapid progress in image processing and computer vision have taken the automation of structural interpretation to a higher level. In this paper, we present a comparison of machine learning algorithms to classify 2D synthetic seismic images correspond to anticlines structures for water and gas, respectively. The reported results established the superiority of CNN compared to other classifiers in terms of classification accuracy.

The structure of this paper is organized as follows: Section 2 describes the steps of the materials and methods, section 3 presents the experimental results and discussion, and the conclusion is given in section 4.

## 2 Materials and Methods

### 2.1 Features and Types of Machine-Learning Algorithms

Various steps are performed on seismic images before the classification process. Initially, a set of features or seismic attributes are selected and forwarded like input to the machine-learning algorithms. Finally, the classifier is used to classify these features and make predictions based in this classification. These steps are used in every experiment of machine learning. It is popular in machine-learning applications to first consider the learning styles that an algorithm can adopt.

*Supervised learning*: Input data is called training data and has a known label or result. A model is prepared through a training process in which it is required to make predictions and is corrected when those predictions are wrong. The training process continues until the model achieves a desired level of accuracy on the training data. Example problems are classification and regression.

*Unsupervised learning*: Input data is not labeled and does not have a known result. A model is prepared by deducing structures present in the input data. This may be to extract general rules. It may be through a mathematical process to systematically reduce redundancy, or it may be to organize data by similarity. Example problems are clustering,
and dimensionality reduction.

### 2.2 The Challenge behind Classification of Seismic Data

Nowadays, in seismic data analysis, seismic interpretation is a critical process. This process aims at identifying structures or environments of significant importance in diverse applications. For example, for oil and gas exploration, a successful interpretation can help identify structures (such as faults, salt domes, and horizons) that are indicators of potential locations of reservoirs. When subsurface structures are the primary interest in seismic interpretation, it is often called structural interpretation [6].

The seismic evaluation produces a large amount of data about structural interpretation. The improvements of these data are essential to mine and process these evaluations efficiently.

Different types and steps of machine-learning algorithms are available for this purpose as shown in Figure 1.

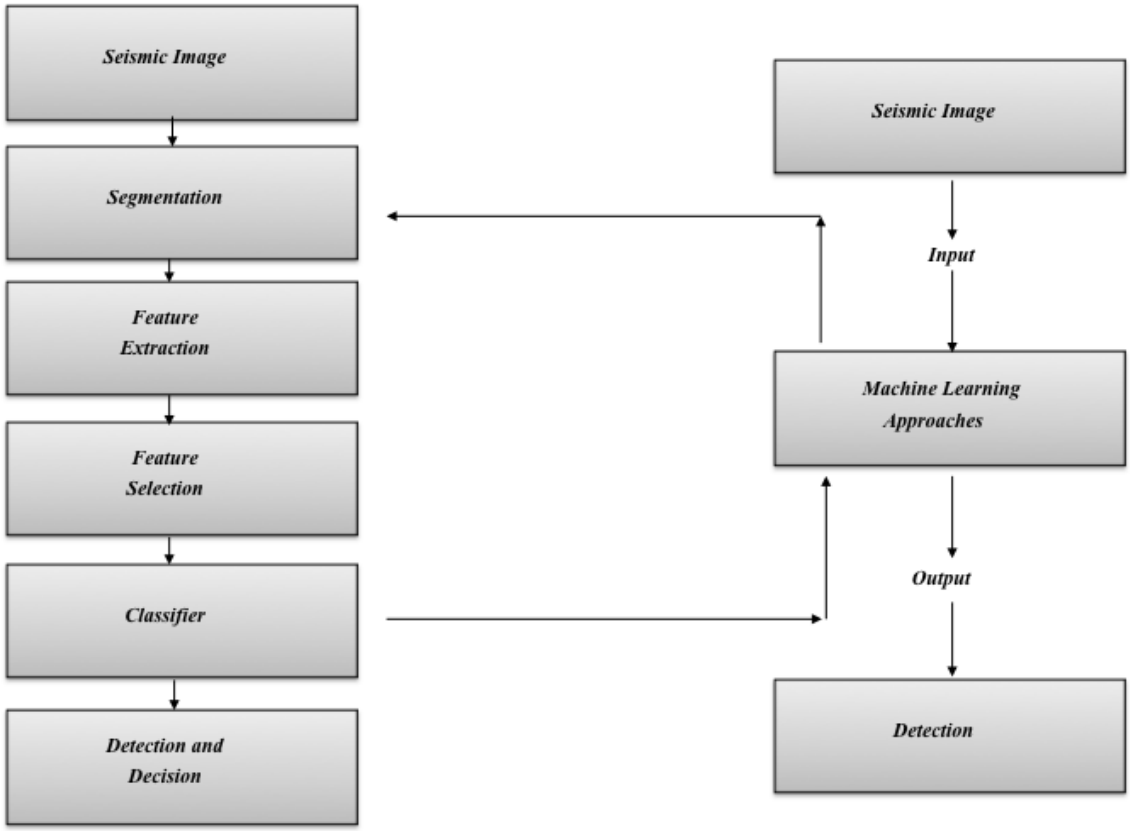


Figure 1. Machine-learning algorithms workflow in seismic images

### 2.3 Automated Classification of Anticlines Structures Using Machine Learning

Our approach aims at automatically identifying geological elements from seismic data; specifically, we focus on the automatic classification of anticlines structures using supervised learning techniques.

In supervised learning we assume each element of study is represented as an n-component vector-valued random variable $(X_1, X_2,..., X_n)$, where each $X_i$ represents an attribute or feature; the space of all possible feature vectors is called the input space $X$. We also consider a set $\{w_1, w_2,..., w_k\}$ 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 $T = (x_i, w_i)$, where $x = (x_1, x_2,..., x_n)$ is a vector in the input space, and $w$ is a value in the (discrete) output space. We 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 $f : X \to W$. We

commonly choose the hypothesis that minimizes the expected value of a loss function (e.g., number of misclassifications).

### 2.4 Classification Methods

Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Machine learning focuses on prediction, based on known properties learned from the training data shown as

*k-nearest neighbor, kNN*

One of the simplest and trivial classifiers is the classifier *k*-nearest neighbor. *kNN* finds a group of *k* objects in the training set that are closest to the test object and bases the assignment of a label on the predominance of a particular class in this neighborhood. There are three key elements of this approach: a set of labeled objects, a distance or similarity metric to compute distance between objects, and the value of *k*, the number of nearest neighbors. To classify an unlabeled object, the distance of this object to the labeled objects is computed, its *k*-nearest neighbors are identified, and the class labels of these nearest neighbors are then used to determine the class label of the object.

*Principal component analysis, PCA*

Principal component analysis (PCA) simplifies the complexity in high-dimensional data while retaining trends and patterns. It does this by transforming the data into fewer dimensions, which act as summaries of features. High-dimensional data are very common in seismic and arise when multiple features or seismic attributes are measured for each sample. This type of data presents several challenges that PCA mitigates; computational expense and an increased error rate due to multiple test correction when testing each feature for association with an outcome. PCA is an unsupervised learning method and is similar to clustering; it finds patterns without reference to prior knowledge. PCA reduces data by geometrically projecting them onto lower dimensions called principal components (PCs), with the goal of finding the best summary of the data using a limited number of PCs. The first PC is chosen to minimize the total distance between the data and their projection onto the PC. By minimizing this distance, we also maximize the variance of the projected points. The second (and subsequent) PCs are selected similarly, with the additional requirement that they be uncorrelated with all previous PCs.

*Random Forest, RF*

Random Forest or random decision forests are an ensemble learning method for regression, classification and other tasks, made from the random selection of samples of the training data. Random features are selected in the induction process. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble. Each tree is grown as described following:

- By sampling *N* randomly, If the number of cases in the training set is *N* but with replacement from the original data, This sample will be used as the training set for growing the tree.

- For *M* number of input variables, the variable *m* is selected such that *m<M* is specified at each node, *m* variables are selected at random out of the *M* and the best split on these *m* is used for splitting the node. During the forest growing, the value of *m* is held constant.
- Each tree is grown to the largest possible extent.

*Support vector machine, SVM*

The support vector machine is an algorithm for data classification and regression. The SVM is an estimation algorithm that separates data in two classes, but since all classification problems can be restricted to the consideration of the two-class classification problem without loss of generality, SVMs can be applied in classification problems in general. SVMS only use information (examples) within the decision borders (called support vectors) and, by means of quadratic programming, they attempt to induce linear or hyperplane separators which maximize the minimum distance between classes. In order to process non-linear ratios, SVM uses kernel functions to project the information in spaces of greater dimensionality and then transform them into linearly separable classes. The selection of an appropriate kernel function is important, since the kernel function defines the feature space in which the training set examples will be classified.

The SVM algorithm is a learning machine; therefore it is based on training, testing, and performance evaluation, which are common steps in every learning procedure. Training involves optimization of a convex cost function where there are no local minima to complicate the learning process. Testing is based on the model evaluation using the support vectors to classify a test dataset. SVM classifier is adopted in this work due to its high accuracy, ability to deal with high-dimensional data, and flexibility in modeling diverse sources of data.

*SVM model parameter selection*

We will consider two parameters:

- The parameter C is a regularization factor and tells the classifier how much we want to avoid misclassifying training examples. A large value of C will try to correctly classify more examples from the training set, but if C is too large it may overfit the data and fail to generalize when classifying new data. If C is too small, then the model will not be good at fitting outliers and will have a large error on the training set.
- The SVM learning algorithm uses a kernel function to compute the distance between feature vectors. In this study we are using a linear base kernel function.

*Convolutional neural networks, CNN*

The concept of deep learning originated from artificial neural network research. Unlike the neural networks of the past, modern deep learning has cracked the code for training stability and generalization and scale on big data. It is often the algorithm of choice for highest predictive accuracy, as deep learning algorithms performs quite well in a number of diverse problems. Deep learning architectures are models of hierarchical feature extraction, typically involving multiple levels of nonlinearity [2]. Among the many supervised learning algorithms currently in use within machine learning, deep-learning algorithms and specifically CNNs occupies a prominent

position [4]. A CNN is a widely used deep-learning technique proven to be state-of-the-art for computer vision problems including image classification, localization, and segmentation. This makes them a good solution for many computer vision tasks.

A CNN processes 2D images to learn representations of data with multiple levels of abstraction. The CNN takes input an image and convolves it with a 2D kernel of adjustable weights. The same kernel is convolved with the input image at different points in the image, which is known as weight sharing technique. Weight sharing reduces the number of free parameters. The results of convolution are added together with an adjustable scalar called a bias. The output is then fed into an activation function, which produces a 2D plane called a feature map. Convolution produces multiple feature maps whose number depends on the number of kernels, which is a function of the architecture. Each feature map is then connected to a subsampling layer, which reduces the size of feature maps. These subsampled feature maps are passed through of an activation function, which helps in retaining nonlinear properties.

As previously mentioned, a convolutional layer takes an input image and convolves it with kernels to produce several two-dimensional planes of neurons called feature maps. Each element of a feature map is obtained by convolving the respective kernel with units in the neighborhood in the previous layer. These outputs obtained after each convolutional layer are then summed up together with a trainable bias term which is then passed to an activation function to obtain each unit of a feature map. Convolutional layers act as feature extractors to extract features, such as corners, edges, endpoints or non-visual features, by convolving the input with kernels consisting of weights [3]. As the weights are shared, the number of parameters to training the neural network is reduced. This also reduces the memory necessary to store these parameters during execution. The convolution operation in each convolutional layer makes a CNN translational and distortion invariant; i.e., when the input image is shifted, the output feature map will be shifted in the same amount as input. The number of kernels in each convolution layer depends upon the number of features maps and varies from architecture to architecture. The network is organized in a hierarchical layer structure that, at each level, combines the lower level features into higher level ones, until the image class label is obtained. The proposed network architecture in this study contains 3 convolutional, activation function ReLU, and max-pooling layers followed by a fully connected layer and ends with a two-class softmax layer. This architecture is summarized in Table 1. What follows is a description of the types of layers:

- Input layer: The input layer has three channels of 32x32 pixels, corresponding to the normalized RGB images.
- Convolutional layers: A convolutional layer convolves the input image with a set of learnable filters, each producing one feature map in its output. The receptive fields (kernels) are of size 3x3, the zero-padding and the stride is set to 1. The three convolutional layers learn 32 feature maps.
- Max-pooling: The lower-level information needs to be spatially integrated for the image region, as well as simplified when accounting for higher level information. Max-pooling layers allow for such a complexity reduction without increasing the number of parameters in the network. The max pooling layers use a stride and pooling size equal to 2.
- Fully connected layers (FC): Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular neural networks. Both the convolutional

layers and fully connected layers are composed of rectified linear units, with activation function f(x) = max(0, x).

- Output layer: The output is composed of two neurons, corresponding to each of the two classes that are normalized with a softmax activation function.

Table 1. CNN architecture

| *Layers* | *1* | *2* | *3* | *4* | *5* |
|---|---|---|---|---|---|
| *Types* | *Co+P* | *Co+P* | *Co+P* | *FC* | *C* |
| *Feature maps* | *32* | *32* | *64* | *128* | *2* |
| *Filer size* | *3x3* | *3x3* | *3x3* | | |
| *Conv. stride* | *1x1* | *1x1* | *1x1* | | |
| *Pooling size* | *2x2* | *2x2* | *2x2* | | |
| *Pooling stride* | *1x1* | *1x1* | *1x1* | | |
| *Padding size* | *1x1* | *1x1* | *1x1* | | |

*Co+P: Convolution and then Pooling, FC: Fully connected, C: Classification.*

### 2.5 Dataset

Two databases of synthetic images are generated:

- First: a database for training, which consists of 400 synthetic images, 200 images correspond to anticlinal structures for water (Class0) and 200 images correspond to anticlinal structures for gas (Class1).
- Second: a database for testing, which consists of 50 synthetic images, 25 images correspond to anticlinal structures for water (Class0) and 25 images correspond to anticlinal structures for gas (Class1).

### 2.6 Training and Testing

The task of training and testing is implemented to evaluate the performance of the CNN based classifier. Here, the dataset along with the class information is used. In our approach, approximately 80% of the entire dataset was used for training, and approximately 20% was used as the validation set. Finally, we then build a classification model that is subsequently used to automatically label a 2D seismic image dataset of testing.

The performance of CNN is improved by adjusting few parameters: batch size and epochs. We stopped the training process after stabilization of the validation accuracy with equal weight for all the classes (100 epochs). The batch size used is 32 samples. The network weights are initialized randomly, and the Adam adaptive learning rate gradient-descent backpropagation algorithm is used for weight updates. The selected loss function is the categorical cross entropy. After training, the trained classifier is validated using the testing dataset and the performance is evaluated in terms of classification accuracy from the resulted confusion matrix [17]. Here, *TP*, *TN* represent true positives and true negatives, which are the diagonal elements of the confusion matrix, and *FP*, *FN* represent the false positives and false negatives, represented by the off-diagonal elements of the resulted confusion matrix: $accuracy = TP + TN / FP + FN$

After completion of the training and testing, the classification performance achieved using this proposed CNN framework has been compared to other supervised classifiers namely *KNN*, Random Forest, and SVM applied to the images transformed by PCA.

(a) (b)

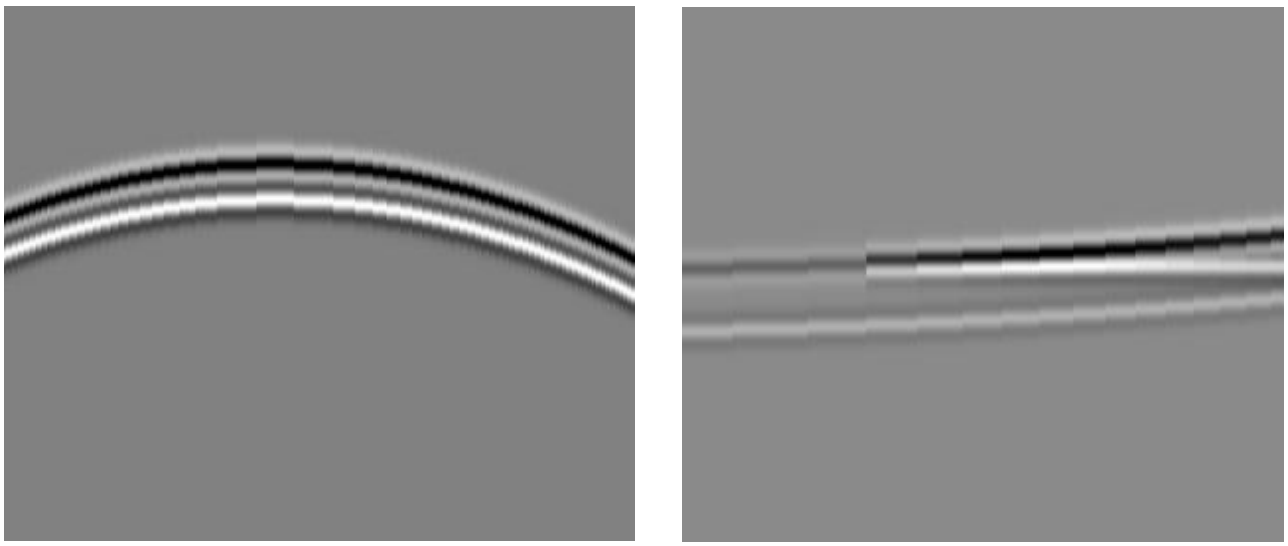

Figure 2. (a) seismic data for water, (b) seismic data for gas. Size image is 320x240 pixels.

### 2.7 Computational Tools

We trained the machine learning algorithms using Scikit-learn [17] and Python 3.6, and the Python deep-learning library Keras 2.0.8 with a TensorFlow 1.3 backend, was used in order to perform the classification through CNN architecture using NVIDIA P100 GPU on Sabine Cluster, at the HPE Data Science Institute, University of Houston.

## 3. Results

The experimental results and analysis carried out in this study are reported in this section.

The machine-learning algorithms PCA, *KNN*, Random Forest, and SVM include a set of parameters that should be optimized in order to achieve good accuracy in the process of

classification. An optimization of some of its parameters was done, and values for default of other parameters for each method are used [17].

- An analysis of PCA was applied to the dataset with the purpose of obtaining seismic attributes sensitive to characterize different textures observed inside an anticlinal structure and in its surroundings. The numbers of PCA were set to 50.
- For *kNN* classification, the numbers of nearest neighbor were set to 3, and Euclidean distance matrix and the nearest rule to decide how to classify the sample were used.
- For Random Forest classification the value of 1000 trees were used.
- In SVM classification, linear kernel function and C = 100 parameters were used to find separating hyperplane.

The optimization of the CNN classifier has been carried out with different combinations of their associated parameters. Table 2 present the results of the comparison in terms of classification accuracy of the machine learning algorithms used in this study. This table shows that our CNN framework proposed yields higher percentage of accuracy when compared to other machine-learning algorithms.

CNN can extract useful and hidden features during training automatically from the original images, which is evidenced by the results obtained in this study.

Table 2. Accuracy for testing data

| | |
|---|---|
| *KNN* | *68%* |
| *SVM* | *92 %* |
| *RF* | *92 %* |
| *CNN* | *96 %* |

In evaluating the effectiveness of our CNN methodology, the confusion matrix is an important measure. Table 3 shows the confusion matrix for the testing data, predicting classes. We can readily see the strong diagonal components. This means that our classifier is achieving little classification error.

Table 3. Confusion matrix

| *Classes* | *Class0* | *Class1* |
|---|---|---|
| *Class0* | *24* | *1* |
| *Class1* | *1* | *24* |

Figure 3 shows the results of the average area under the ROC curve (AUC) of the 2 classes. We

can see that the curve follows the left-hand border and then the top border of the ROC space, showing the predictive model has high precision.

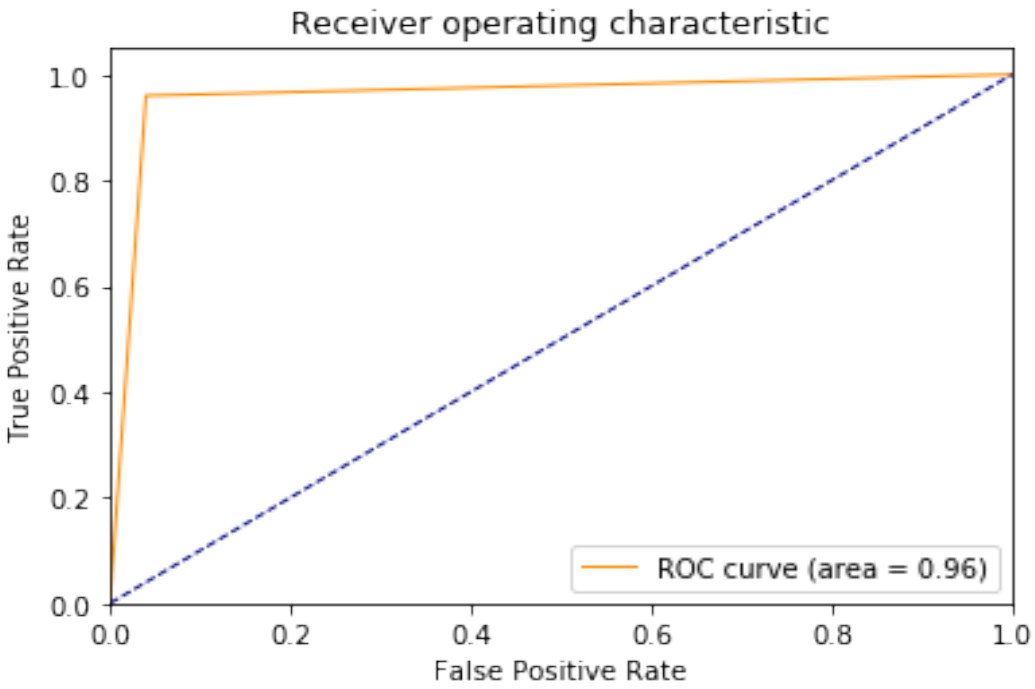


Figure 3. Area under the ROC curve

Therefore, it can be inferred from the results that the proposed framework based on CNN could be used as a powerful tool to classify anticlines structures from seismic data.

## 4. Conclusion

In this work, a framework based on CNN is proposed to classify anticlines structures from seismic data into two classes, water and gas, respectively. Comparative analysis reported in this study has established the superiority of the proposed methodology compared to other supervised classification algorithms in terms of accuracy. The performance of the CNN classifier considering different architectures and associated parameters has also been investigated.

The proposed work signifies the applicability of the concepts of image processing and machine learning to structural interpretation, thus expanding the scope of interdisciplinary researches.

We conclude that CNN is a promising mechanism to identify geological structures on seismic data. We ascribe the efficiency of CNN to the capacity to model complex decision boundaries needed during class discrimination. Finally, this study provides some evidence that using machine-learning techniques, as deep learning, is a promising mechanism for seismic structural evaluation.

## Acknowledgments

The author thanks Dr. C. Cobos from Reposal Company for providing the images to present the experimental results. This work was supported by the HPE Data Science Institute.

## References

[1] Valiant, L., G. A theory of the learnable. Commun. ACM, 27(11): 1134–1142, November 1984.

[2] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1). Cambridge: MIT Press.

[3] Schmidhuber, J., Deep learning in neural networks: An overview. Neural networks, 2015. 61: p. 85-117.

[4] LeCun Y., Kavukcuoglu, K., and Farabet, C. "Convolutional networks and applications in vision" in Circuits and Syst. (ISCAS), Proc. of the 2010 IEEE Int. Symp., pp. 253–256.

[5] Zheng, Y., Zhang, Q., Yusifov, A., Shi, Y. Applications of supervised deep learning for seismic interpretation and inversion. The Leading Edge, Special Section: Machine learning applications, July 2019, Volume 38, No. 7, 526-533.

[6] AlRegib, G., Deriche, M., Long, Z., Di H., Wang, Z., Alaudah, Y., Shafiq, M., and Alfarraj, M. Subsurface structure analysis using computational interpretation and learning: IEEE Signal Processing Magazine, 2018, 35, no. 2, 82–98

[7] Amin A., Deriche M., Hegazy T., Wang Z, AlRegib G. 2015. A Novel Approach for Salt Dome Detection using a Dictionary-based Classifier. SEG 2015, 85th Annual Meeting, New Orleans.

[8] Araya-Polo, M., Dahlke, T., Frogner, C., Zhang, C., Poggio, T., Hohl, D. Automated fault detection without seismic processing: The Leading Edge, 2017, 36, no. 3, 208–214.

[9] Guillen, P., Larrazabal G., Gonzalez G., Boumber D., Vilalta R. 2015. Supervised learning to detect salt body. SEG 2015, 85th Annual Meeting, New Orleans.

[10] Shafiq M., Wang, Z., Amin A., Hegazy T., Deriche M., AlRegib G. 2015. Detection of Salt-dome Boundary Surfaces in Migrated Seismic Volumes Using Gradient of Textures. SEG 2015, 85th Annual Meeting, New Orleans.

[11] Shafiq, M., Alshawi T., Long, Z., and AlRegib, G., The role of visual saliency in the automation of seismic interpretation. Geophysical Prospecting, 66, no. S1, 132–143.

[12] Strigl, D., Kofler, K., and Podlipnig, S. "Performance and scalability of GPU- based convolutional neural networks," in Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro Int. Conf. IEEE, 2010, pp. 317–324.

[13] Waldelan, A., Solberg, A., 2017. Salt Classification using Deep Learning. Proceedings of 79th EAGE Conference and Exhibition, Amsterdam

[14] Wei X., Xu J., Yue M., Wang, Y., Nasher M., Mustafa N., and Yi L. Seismic fault detection with convolutional neural network. GEOPHYSICS, VOL. 83, NO. 5, 2018; P. O97–O103.

[15] Zhang, C., Frogner, C., Araya-Polo, M., Hohl, D. 2014. Machine Learning Based Automated Fault Detection in Seismic Traces. Proceedings of 76th EAGE Conference and Exhibition, Amsterdam.

[16] Zhen, W., Haibin, D., Shafiq, M., Alaudah, Y., AlRegib, G. Successful leveraging of image processing and machine learning in seismic structural interpretation: A review. The Leading Edge, 2018, 451–461.

[17] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

## Biographical information

**Dr. PABLO GUILLEN RONDON** is a research assistant professor/faculty at the HPE Data Science Institute, University of Houston, Houston, TX. He holds a BSc degree in Mathematics and an MSc degree in Applied Mathematics, a DAS. in

Automation Engineering, a PhD in Biomedical Engineering, and a postdoc in Computational Science from University of Texas at El Paso. During the last 20 years he has been working on research and development projects related to oil, gas, and biomedical sciences. These projects have been in different areas such as artificial intelligence, machine learning, data mining, data reconciliation and fault diagnosis, reservoir simulation, geophysics, geothermal, gas, processing of signals, images, and visualization. Universities and companies funded these projects. He has been a keynote speaker at several conferences, and he has over 115 papers published in journals and conference proceedings.