

A Deep Transfer Learning Framework for Seismic Data Analysis: A Case Study on Bright Spot Detection

Julia El Zini, *Member, IEEE*, Yara Rizk, *Member, IEEE*, and Mariette Awad, *Member, IEEE*

Abstract—Bright spots, strong indicators of the existence of hydrocarbon accumulations, have been primarily used by geophysicists in oil and gas exploration. Recently, machine learning algorithms, adopted to automate bright spot detection, have mainly relied on feature extraction and shallow classification workflows to achieve an 85.4% F1 score at best, on 2D seismic data. Deep neural networks have proved their effectiveness in image classification applications, outperforming humans in some instances, but have not been applied to bright spot detection yet. However, their data-hungry nature poses a challenge in domains suffering from expensive data acquisition such as seismic data analysis problems; they generally require millions of training samples before achieving good performances. In this work, we implement *SeisNet*, a convolutional neural network with a “butterfly” architecture that overcame the limited data challenge by implementing data augmentation and inductive transfer learning techniques. Moreover, we adopt a novel formulation that allows us to detect bright spots and estimate their volume. Our approach was tested against different pretraining and transfer learning methods and was shown to outperform other approaches in the literature by achieving a 95.6% F1 score on bright spot detection. Our model accurately predicted the volume of the bright spot with an average absolute error that is no more than 0.04% of the total volume of the seismic image. This work is an important step in establishing pretrained networks for other seismic applications such as earthquake prediction; our domain-specific pretrained network, proven to outperform state-of-the-art pretrained networks on bright spot detection, may be used to jump-start the training of deep models on other seismic problems.

Index Terms—Bright spot detection, Seismic analysis, Deep learning, Convolutional neural networks, Transfer learning

I. INTRODUCTION

Oil and gas are still the primary energy source in the world. Internationally, governments spend between 775 billion dollars to 1 trillion dollars annually, to support the oil and gas industries¹. As reservoirs become depleted, the discovery of new sources is essential to sustaining our current lifestyles until alternate, renewable sources of energy become viable, efficient, large scale energy suppliers. The identification, characterization and monitoring of oil and gas reservoirs is a crucial phase of oil and gas extraction, when companies spend billions of dollars [1] without making any profit for almost a decade [2]. Inaccurate interpretation of the data

J. El Zini, Y. Rizk and M. Awad are with the Department of Electrical and Computer Engineering, American University of Beirut, Beirut, Lebanon, e-mail: {jwe04,yar01,mariette.awad}@aub.edu.lb.

¹<http://priceofoil.org/fossil-fuel-subsidies/>

can result in incorrect decisions that would cost oil and gas companies billions of dollars in losses. Reducing the amount of wasted resources on inefficient oil and gas retrieval solutions can allow the reallocation of the saved money to improving renewable energy solutions.

Seismic reflection is one of the most popular techniques used to locate hydrocarbon accumulations by recording the waves reflected from sub-surface layers after projecting a seismic wave toward the earth [3]. This produces a very noisy image of the hydrocarbon underneath Earth's surface and must be processed before accurate interpretation of the data can occur. Failing to accurately interpret this image can lead to incorrect identification of oil and gas reservoirs.

Different methods have been investigated for noise reduction in seismic data from digital filtering [4] to various optimization techniques [5]. Recently, machine learning algorithms, and artificial neural networks (ANN) specifically, have proven to be effective in processing noisy data and extracting meaningful features with a low error rate. For seismic data analysis, ANN have achieved high accuracies in predicting diverse physical and geometrical properties of subsurface geological structures [6], [7]. Convolutional neural networks (CNN), a variant of ANN formulated to process images, have been successfully applied to many image processing problems including image classification [8], localization [9], [10], segmentation [11], and annotation [12], [13], not only for 2D data but also for 3D data.

Deep learning has been adopted in some seismic data analysis applications such as earthquake detection [14]. However, bright spot detection using deep learning may be a challenge because large amounts of labeled data are required for effective learning. Labeling seismic data is expensive; it requires experts to sift through millions and millions of data points with only a few of them containing bright spots leading to unbalanced datasets. Unbalanced data also poses a challenge to deep learning approaches that may not have enough data from a given class to learn an accurate model.

In this work, we propose to apply CNN to bright spot detection from seismic data by adopting a semi-supervised learning approach. Data augmentation and pretraining CNN using unlabeled seismic data are adopted to combat the limited and expensive data labeling processing. Then, the resultant model is fine-tuned in a supervised fashion using a small subset of labeled images. Our approach is compared to other works in the literature. Results reveal the importance of pretraining on seismic data as opposed to datasets from other

domains (e.g. handwriting recognition's MNIST or object recognition's Imagenet). Furthermore, domain-specific deep learning techniques proved more effective than other feature extraction based machine learning algorithms. In summary, our approach outperformed other methods in the literature by achieving an F1 score of 95.6%, a 1.4% improvement over the next best model.

Next, section II briefly surveys existing machine learning and deep learning approaches to seismic data analysis and bright spot detection, specifically. Then, section III details the proposed methodology. Section IV describes the experimental setup adopted in this work before section V reports on the empirical results. Finally, section VI concludes with final remarks and future work.

II. LITERATURE REVIEW

A. Deep Learning for Seismic Data Analysis

Deep learning has been applied to many real-world problems from natural language understanding and generation to scene understanding and speech recognition in various domains such as healthcare, transportation, remote sensing [15] and geological sciences. Focusing on seismic data analysis, seismic data consist of recordings obtained using the seismic reflection method to characterize sub-surface composition [3]. This approach emits high energy sound waves which traverse through various sub-surface media before reemerging to the surface. Receivers record the output and generate noisy signatures. A wide range of signal processing techniques is required to reconstruct the various layers beneath the Earth's crust and identify its characteristics including porosity, fluid saturation and bright spot existence. Deep neural networks have attempted to solve various geological problems including seismic velocity model building [16], predicting fault probabilities, strikes, and dip [17], seismic pattern recognition and classification [18] and weak signal preservation [19].

Specifically, CNN have been trained to identify salt-body boundaries from seismic data, outperforming shallow learning algorithms [20]. Zeng et al. adopted the U-net and ResNet CNN architectures to train salt-body interpretation models [21]. Recurrent neural networks were trained on hyperspectral images by treating hyperspectral pixels as sequential data and achieved comparable accuracies to state-of-the-art approaches [22]. Hunag et al. efficiently identified geological features from seismic attributes with high accuracy by combining CNN and shallow machine learning algorithms [23]. Their results showed that they accurately identified faults with a scalable workflow. Deep belief networks and stacked auto-encoders, two types of deep neural networks, were also trained to identify seven seismic events from volcano-seismic data; they outperformed and converged faster than classical machine learning algorithms [24].

Araya et al. adopted a data driven approach to eliminate the expensive physical modeling step by training deep neural networks on raw seismic data to automate fault detection [25] and reconstruct velocity models [26], both essential phases in hydrocarbon exploration. For the former model, they used the Wasserstein loss function to handle the spatial layout

dependency of the output and validated their approach on synthetic datasets, reaching an area under the ROC curve (AUC) value of 0.919 [25]. For the latter, they learned the tomography operator using deep neural networks from raw seismic data to generate velocity models; experiments on synthetic data revealed promising results [26]. Perol et al. proposed a scalable CNN, termed ConvNetQuake, for earthquake detection and localization that needed only one waveform and achieved a prediction rate 17 times better than other approaches [14]. ConvNetQuake processed the windowed three-channel waveform in approximately 1 minute, predicted whether the signal is noise or an actual seismic event, and provided the geographical location of the identified event. Kanarachos et al. predicted the occurrence of earthquakes by identifying anomalies in seismic signals [27]. To that end, they proposed a deep learning workflow that first applied a wavelet transform to the raw seismic data, trained deep temporal neural networks then applied the Hilbert transform on the network's output to achieve a prediction accuracy of 86% [27].

The butterfly network architecture, also known as the encoder-decoder architecture, is used by Shi et al. to automatically capture subtle salt features from the 3D seismic images [28]. This architecture was shown to achieve a 96% accuracy when trained on SEAM Phase I data set [29]. Moreover, Pham et al. utilized the encoder-decoder CNN to solve the automatic channel detection problem in seismic images [30]. Their approach relies on training a network on synthetic data and fine-tuning it on real field data and achieved a 95.6% as a mean value for intersection over union. The same encoder-decoder CNN architecture was used for facies classification in [31]. The CNN was trained on the F3 seismic survey acquired in the North Sea, offshore Netherlands and achieved a 94.1% accuracy. Recently, Wu et al. trained a butterfly convolutional neural network on synthetic seismic images and their corresponding binary fault labeling images to develop an end-to-end fault segmentation network that achieves a pixel accuracy of 95% [32].

All the above-mentioned references trained deep networks from scratch using big data. However, in some situations, seismic data is not available in abundance; labeling is expensive and time constraints may be imposed on data collection. Some work in the literature has proposed frameworks that do not require significant amounts of data. One approach investigated transfer learning for earthquake prediction from seismic data by using a CNN pretrained on MNIST then fine tuning the model using a small set of labeled seismic data [33]. An accuracy of 96.8% over four seismic stations was achieved. Peters et al. tackled the seismic horizon tracking problem with limited data by formulating the problem as a regression instead of classification problem [34]. This allowed them to learn from limited data while achieving competitive accuracies.

B. Bright Spot Detection

Multiple machine learning approaches have been investigated for bright spot detection, an essential step in hydrocarbon exploration. Various feature extraction techniques have been derived to identify bright spots. Extracting statistical and

TABLE I: Nomenclature

Symbol	Definition
$\mathbf{x}^{(k)} \in \mathbb{R}^{m \times n}$	k^{th} input image
$\mathbf{y}^{(k)} \in \mathcal{Y} \subseteq \{0, 1\}^{m \times n}$	Output label matrix corresponding to image $\mathbf{x}^{(k)}$
N	Number of labeled samples
M	Number of unlabeled samples
$\mathbf{x}_l^{(i)} \in \mathbb{R}^{m \times n}$	i^{th} labeled image
$\mathbf{x}_u^{(i)} \in \mathbb{R}^{m \times n}$	i^{th} unlabeled image
\mathcal{T}_S	Source task of transfer learning
\mathcal{D}_S	Source domain of transfer learning
\mathcal{X}_S	Space of images in the source domain
X_S	Training sample in the source domain
$P(X_S)$	Probability distribution of X_S
\mathcal{T}_T	Target task of transfer learning
\mathcal{D}_T	Target domain of transfer learning
\mathcal{X}_T	Space of images in the target domain
X_T	Training sample of the target domain
$P(X_T)$	Probability distribution of X_T
$y_{cl}^{(k)} \in \{0, 1\}$	Actual output of bright spot classification for image k
$\hat{y}_{cl}^{(k)} \in \{0, 1\}$	Predicted output of bright spot classification for image k
$y_{vol}^{(k)} \in \mathbb{R}$	Actual bright spot volume in image k
$\hat{y}_{vol}^{(k)} \in \mathbb{R}$	Predicted bright spot volume in image k
$\hat{\mathbf{y}}^{(k)} \in \mathcal{Y} \subseteq \{0, 1\}^{m \times n}$	Predicted $m \times n$ label matrix consisting of binary input (after thresholding)
$\hat{\mathbf{y}}_{pre}^{(k)} \in \mathbb{R}^{m \times n}$	$m \times n$ matrix outputted by our model (before thresholding)
C_1, C_2	Thresholds

texture features has been at the core of these approaches. Chopra et al. [35] computed gray-level co-occurrence matrices and Haralick features to predict the existence of oil reservoirs. They also computed second order statistical measures such as energy, contrast and entropy that allowed them to characterize the distribution and connectivity of these reservoirs. Rizk et al. [6] proposed to apply the continuous wavelet transform to seismic images before extracting statistical and texture features. Then, they trained shallow classifiers such as support vector machines, k-nearest neighbor and shallow ANN on labeled data to achieve an f-measure of 91.5% using ANN. To the best of our knowledge, none of the references in the literature adopted a deep learning or transfer learning workflow to identify bright spots from seismic data.

III. METHODOLOGY

We first formulate the bright spot detection problem. Then, we describe the two approaches adopted to overcome the limited data problem: data augmentation and transfer learning. Finally, we describe the overall workflow of our proposed framework which includes the unsupervised training of a CNN model using big data and the supervised fine-tuning of this model using a small set of labeled data. Table I summarizes the adopted nomenclature in the remainder of this paper.

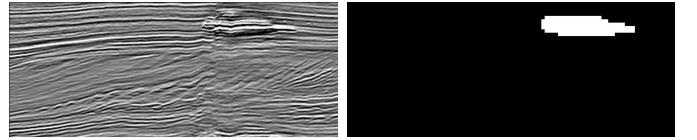


Fig. 1: Left: Input example. Right: Corresponding output.

A. Problem Formulation

In their work, Rizk et al. [6] did not train their supervised learning models on the seismic images but rather on 8×8 windowed sub-images. Each sub-image was labeled as 1 if the majority of its pixels were annotated as having a bright spot and 0 otherwise. The sub-images were then fed independently to the learning model i.e. neighboring sub-images were not taken into consideration in the classification process. Knowing that bright spots are not usually scattered over the surface but assembled in one area, one can see the limitations of this formulation as it fails to capture the correlation between having bright spots in neighboring windows.

In this work, we address the aforementioned problem by considering the whole seismic image as one training instance. In addition to detecting the presence of a bright spot, we build a model that can predict its volume and localize it in the image. The problem is no longer a binary classification; the output is formulated as a two-dimensional logical matrix having the same size as the input image, as shown in Fig. 1. For each $n \times m$ input image $\mathbf{x}^{(k)}$, (1) defines the output label $\mathbf{y}^{(k)}$ as a $n \times m$ binary matrix.

$$\mathbf{y}^{(k)}[i, j] = \begin{cases} 1 & \text{if pixel } (i, j) \text{ is labeled as a bright spot} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Once $\mathbf{y}^{(k)}$ is estimated, the volume of the bright spot can be easily computed if the mapping between the dimension of the images and the geographical coordinates is known. This mapping is provided in the standard data formats [36] and the seismic analysis tools.

This labeling technique introduces two challenges. First, the input and output dimensions increase with a factor of $\frac{n}{w} \times \frac{m}{w}$ which is significant for large n and m . As a result, shallow classifiers would perform poorly on our data which is high-dimensional in both the input and the output. This implies that deep learning must be adopted. However, it requires large amounts of labeled data to learn accurate models. This leads to the second challenge. Not only is collecting and labeling seismic data expensive, but our formulation will reduce the number of training instances by a factor of approximately w^2 . With only 110 labeled images [6] at our disposal, data-hungry deep models would fail on such a small dataset. Next, we discuss two adopted solutions to overcome these challenges: 1) data augmentation and 2) transfer learning.

B. Data Augmentation

Synthetic data augmentation generates more data from limited datasets and aids in preventing over-fitting during

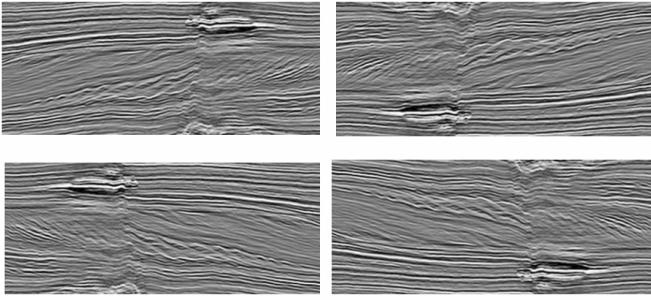


Fig. 2: Sample seismic image (top-left) and its rotated (top-right), horizontally flipped (bottom-left) and vertically flipped (bottom-right) versions.

the training of deep networks [37]. Different augmentation techniques are used in the literature to increase the size of the training dataset such as translation, rotation, adding noise, flipping and cropping [38], [39]. To maintain the semantic correctness of the augmented images, we adopted two techniques: rotation of 180° and flipping, as shown in Fig. 2. Although vertical flipping might produce images that could rarely be encountered in real-life seismic conditions, it could be thought of as a noise or distortion at training time that can help strengthening the model.

C. Transfer Learning

During training, deep networks first learn non-linear representations of the input [40] as low-level and high-level features, then learn a classification from the learned features. When limited labeled data is available, transfer learning can relay knowledge from the source domain (where deep representations are learned on large sets of unlabeled data), to the target domain (where the classification is learned on these representations). Specifically, we cast our problem as an instance of inductive transfer learning where labeled data is available in the target domain but not in the source domain [41].

For instance, we are given a set of N labeled training samples

$$\{(\mathbf{x}_l^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}_l^{(2)}, \mathbf{y}^{(2)}), \dots, (\mathbf{x}_l^{(N)}, \mathbf{y}^{(N)})\}$$

where each $\mathbf{x}_l^{(i)}$ is a vector of pixels intensities representing the image i (having a subscript “ l ” to indicated that it is a labeled instance) and $\mathbf{y}^{(i)}$ is the label vector defined as in (1). In addition, we are given a set of unlabeled data

$$\mathbf{x}_u^{(1)}, \mathbf{x}_u^{(2)}, \dots, \mathbf{x}_u^{(M)}$$

having the same modality as $\mathbf{x}_l^{(i)}$, i.e. images, but not necessarily drawn from the same probability distribution.

We define $\mathcal{D}_S = \{\mathcal{X}_S, P(X_S)\}$ to be the *source* domain of our transfer learning problem, with \mathcal{D}_S consisting of unlabeled seismic images \mathcal{X}_S along with their probability distribution $P(X_S)$ where $X_S = \{\mathbf{x}_u^{(1)}, \mathbf{x}_u^{(2)}, \dots, \mathbf{x}_u^{(M)}\} \in \mathcal{X}_S$ is a particular unlabeled learning sample. The *task* \mathcal{T}_S is to learn a mapping $f_S : \mathcal{X}_S \mapsto \mathcal{F}$ where \mathcal{F} is the feature space that best represents the input \mathcal{X} . We further define $\mathcal{D}_T = \{X_T, P(X_T)\}$

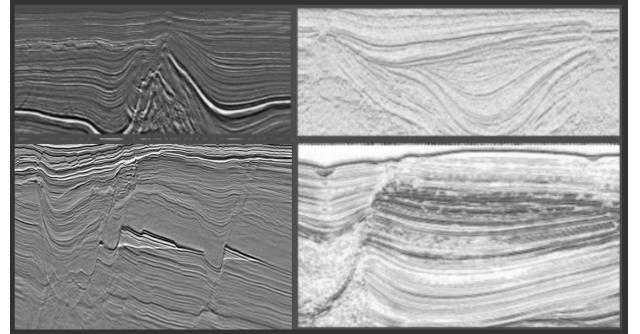


Fig. 3: Samples of seismic images used in the training of the source task.

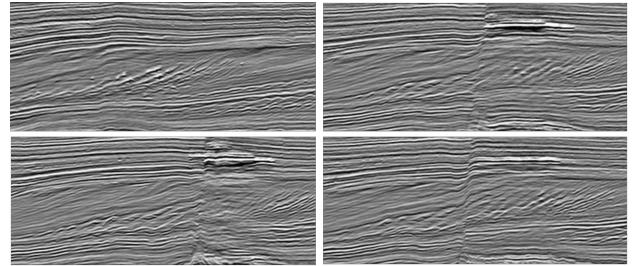


Fig. 4: Samples from the labeled dataset used in the training of the target task.

to be the *target* domain, with \mathcal{D}_T consisting of labeled seismic images \mathcal{X}_T along with their probability distribution $P(X_T)$ where $X_T = \{\mathbf{x}_l^{(1)}, \mathbf{x}_l^{(2)}, \dots, \mathbf{x}_l^{(N)}\} \in \mathcal{X}_T$ is a particular labeled learning sample. The *task* \mathcal{T}_T is to learn a mapping $f_T : \mathcal{X}_T \mapsto \mathcal{Y}$ where \mathcal{Y} is the label space.

Under these definitions, transfer learning aims at improving the performance of \mathcal{T}_T , when $\mathcal{T}_S \neq \mathcal{T}_T$ or $\mathcal{D}_S \neq \mathcal{D}_T$. Clearly, the former is satisfied since \mathcal{T}_S is to learn image representations whereas \mathcal{T}_T is to learn the bright-spot classification, but we also argue that the latter holds for this problem. In fact, the probability distributions in both tasks are different, which leads to two different domains: \mathcal{D}_S and \mathcal{D}_T . Fig. 3 shows different samples of seismic images in the source domain, whereas Fig. 4 shows samples from the dataset of [6] used in the target domain. One can clearly see that the probability distributions differ from the source to the target domain since the data in [6] does not generalize well to different seismic images.

D. Transfer Learning Formulation for SeisNet

We further detail the source and target tasks within the seismic framework and we explain how the transfer from the former to the latter is achieved.

1) Source task: The source task is to learn feature representation of seismic images. Autoencoders [42] are successful in such tasks as they learn to compress data from the input space into a shorter code, and then uncompress the learned code into something that is similar to the original data in an unsupervised manner. In this work, we adopted the autoencoder with the architecture illustrated in Fig. 5 to learn the source task. The

network is referred to as *SeisNetPre* in this paper as it will serve as a pretrained network for this seismic application.

2) *Target task*: The target task is to predict the $n \times m$ $\hat{\mathbf{y}}$ matrix given a $n \times m$ seismic image. The same architecture of the deep network adopted in \mathcal{T}_S is used in the target task with the output of the network being the labels matrix rather than the input image itself. The main reason behind adopting this “butterfly” architecture is to mimic the way the source task works: the first half of the network will be in charge of learning a representation of the image in the context of bright spot detection, whereas the second half will build the label matrix on top of this representation while mirroring the input image. The network is called *SeisNet* throughout this paper.

3) *From the source to the target task*: Since both tasks will first learn a representation of the data, the learned knowledge in the first part of the source task’s network can be transferred to the target task. Fig. 6 illustrates how the weights learned in the first half of the “butterfly” *SeisNetPre* network are transferred to its analogue in *SeisNet*.

Various scenarios [41], [43], [44] could be used to transfer the learned features. The first approach uses the network trained on the source domain data as a feature extractor i.e. the layers of the second part are removed and the rest, with fixed weights, are used to extract a feature vector for the dataset of the target domain. This feature vector will be the input of the network in the target domain. In this work, this is achieved by initializing the weights of *SeisNet* to those of *SeisNetPre* while keeping the weights of the first half of the network frozen. The second approach fine tunes the network learned on the source task on the data of the target task. In this work, this is achieved by initializing the weights of *SeisNet* to those of *SeisNetPre* and fine tune them by backpropagation on the labeled data.

E. Overall Workflow

Fig. 7 illustrates the overall workflow of our approach. Data augmentation is applied to both the labeled and the unlabeled data. The unlabeled data is fed to an autoencoder that learns the weights of *SeisNetPre*. The learned knowledge is transferred to *SeisNet* that is trained on the labeled augmented data to output a label matrix. This matrix is then processed to predict the presence of a bright spot, its location and its volume.

IV. EXPERIMENTAL SETUP

A. Hardware and Software

The experiments were run on an Intel Xeon 64-bit 12-core processor machine with a Quadro K2000 NVIDIA GPU. The algorithms were written in Python 2.7 using the Keras and Tensorflow packages. A 5-fold cross validation was adopted in all experiments and the reported accuracies were averaged over all folds.

B. Data Acquisition

1) *Unlabeled Data*: The unlabeled data on which *SeisNetPre* was trained, is a subset of a 7TB seismic dataset². The

²Accessible through <https://www.equinor.com/en/news/14jun2018-disclosing-volve-data.html>

data was originally presented in the Seg-Y format and was processed to produce around 1K grayscale training images. The training images are all migrated seismic images and are further normalized before being inputted to *SeisNetPre*.

2) *Labeled Data*: The data used in the training of *SeisNet* is adopted from [6] and consists of 110 grayscale images of size 4017×1690 . All images are normalized and data augmentation was applied to all images in the dataset and resulted in a larger dataset of size 440. The data is balanced with a 1 to 0 ratio of 9:22.

C. Evaluation Metrics

To test the ability of our bright spot prediction and volume estimation models, we evaluate our model against two types of metrics: classification and regression.

1) *Classification*: The goal is to predict $\hat{y}_{cl}^{(k)}$ for image k , where $\hat{y}_{cl}^{(k)} = 1$ if image k has a bright spot and 0 otherwise. Given that our model does not have a simple binary output but rather a binary image, a mapping is needed from the predicted output to 0 or 1. Knowing that the bright spot volume cannot be insignificant, one can count the number of pixels P classified as having bright spots and compare this number to a certain threshold C_1 . The image is classified as having a bright spot if $P > C_1$ as in (2). For this task, we report the accuracy and the F1 score.

$$\hat{y}_{cl}^{(k)} = \begin{cases} 1 & \sum_{i=1}^n \sum_{j=1}^m \hat{y}^{(k)}[i, j] \geq C_1 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

2) *Regression*: $\hat{y}_{vol}^{(k)}$ in image k is estimated as a count of the pixels predicted as 1 as in (3).

$$\hat{y}_{vol} = \sum_{i=1}^n \sum_{j=1}^m \hat{y}^{(k)}[i, j] \quad (3)$$

It should be noted that the last layer of the model is a softmax which leads to a $n \times m$ output matrix \hat{y}_{pre} consisting of real values between 0 and 1. A thresholding step is needed to map \hat{y}_{pre} into \hat{y} which consists only of 0s and 1s using (4).

$$\hat{y}^{(k)}[i, j] = \begin{cases} 1 & \hat{y}_{pre}^{(k)}[i, j] > 1 - C_2 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

V. EXPERIMENTAL RESULTS

In what follows, we first test the importance of the data augmentation and the transfer learning techniques used in this work. Then, we study the effect of using domain-specific pretrained network in the source domain of the transfer learning and the effect of using fine-tuning or feature extractor as transfer methods. We also test the effect of the threshold used in (4) on the performance of different pretrained and we compare our results to [6]. Finally, we further improve our results by processing the output matrix with a morphological opening and we study its effect on the performance.

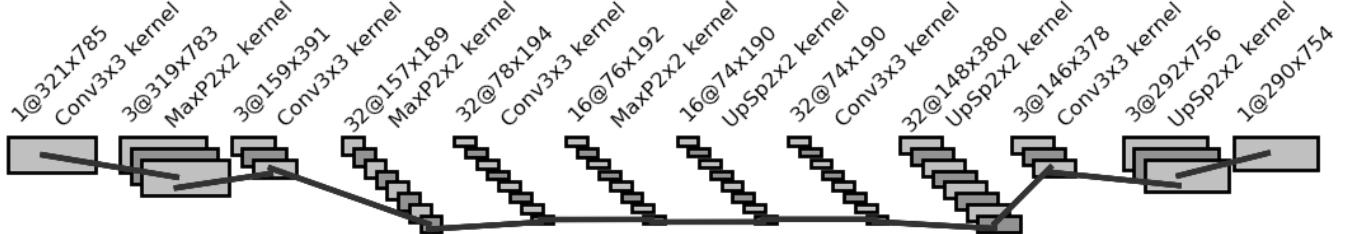


Fig. 5: Autoencoder network architecture (Conv, MaxP and UpSp stand for Convolutional, MaxPooling and UpSampling layers respectively).

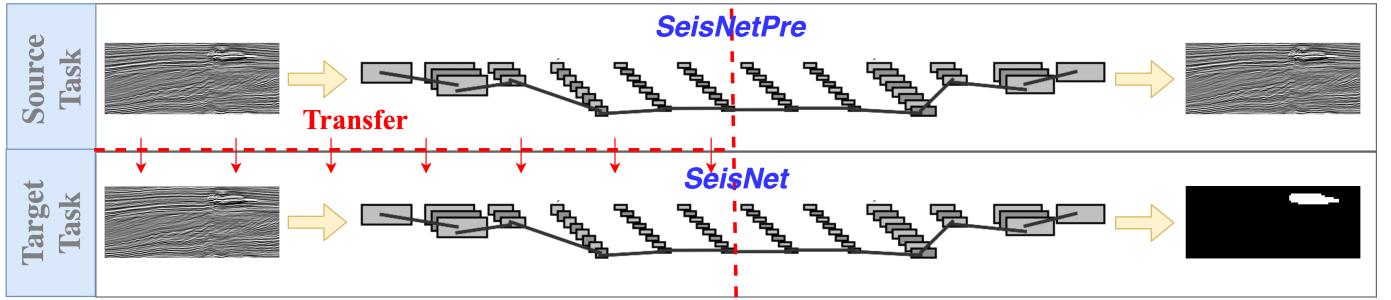


Fig. 6: Transfer between the source and the target tasks

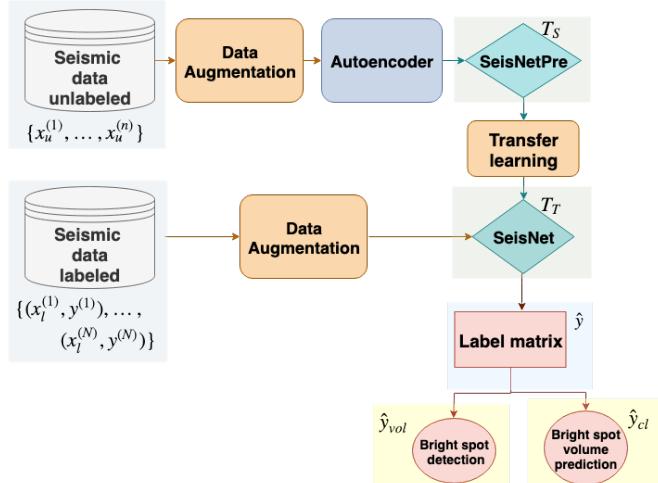


Fig. 7: Overall Workflow

A. Data Augmentation Effect

We studied the effect of data augmentation on the performance of *SeisNet* by training *SeisNet* on the original non-augmented data of size 110. The results are reported in Table II. Clearly, data augmentation improved the results drastically from an F1 score of 31.6% to 79.0% when the transfer learning is not applied and from an F1 score of 46.8% to 95.6% when *SeisNetPre* is used as a pretrained network. Table II shows that the accuracy (Acc), mean absolute error (MAE), mean squared error (MSE), mean squared log error (MSLE), median absolute error (MedAE) and percentage error (PErr) all improved with data augmentation.

B. Transfer Learning Performance

To evaluate the performance of transfer learning, we trained *SeisNet* with randomly initialized weights, i.e. without transferring knowledge from the source task. The regression and classification performances are compared to those of *SeisNet* when initialized with the weights of *SeisNetPre*. Table II shows that pretraining substantially improved the performance as the F1 score increased from 36.8% to 46.8% without data augmentation and from 66% to 95.6% when the augmentation is applied on the labeled data. Similarly, other metrics improved the pretrained network was adopted. When transfer learning is applied in Table II, the transfer method is fixed for both models to the fine-tuning approach. Grid search was applied to choose the threshold C_2 of (4) and the corresponding results were reported .

C. *SeisNetPre* Performance as a Source Task

To evaluate the importance of using the domain specific network *SeisNetPre* as a source task, we tested different pretrained networks: VGG [10], ResNet [45] and a CNN trained on MNIST dataset [46] as a source task. MNIST pretraining is introduced to allow for a fair comparison with the work in [33]. Both approaches explained in section III-C are used to transfer knowledge from the source task to the target task: 1) use the networks as features extractors with fixed weights and 2) fine-tune the weights on the target task. The networks are trained until convergence, i.e. until the loss does not change. Training when *SeisNet* is used as a feature extractor took 186 epochs; whereas when ResNet VGG and MNIST are used, training took 187, 142 and 103 epochs respectively. Additionally, when *SeisNet* is finetuned,

Augmentation	Transfer	Classification		Regression				
		Acc (%)	F1 (%)	MAE	MSE	MSLE	MdAE	PErr (%)
None	None	31.60	36.81	13,604	2.9E12	34	554,098	5.44
None	<i>SeisNetPre</i>	41.71	46.82	3,604	3.8E08	25	67,132	1.44
Rotation & Flipping	None	79.00	66.01	2,540	1.9E07	16	2,730	1.01
Rotation & Flipping	<i>SeisNetPre</i>	91.95	95.60	503	1.8E06	3	814	0.19

TABLE II: Effect of data augmentation and transfer learning on the performance of *SeisNet*

Fig. 8: Samples of training dataset for VGG (left), ResNet (middle) and MNIST (right)

the training needed 187 epochs to converge while ResNet VGG and MNIST needed 152, 146 and 101 epochs respectively.

Table III reports the results of testing *SeisNet* while changing the following factors: the pretrained network, the transfer method and the threshold C_2 . For each network, the threshold is changed and the best results achieved on the F1 score are highlighted in gray. If two models gave the same F1 score for two different thresholds, the one that minimizes the MSE is selected and if both have the same MSE the model with the least threshold is used to avoid overfitting. Moreover, bold font is used to highlight the best results achieved across all models and all thresholds for every reported metric.

Table III shows that pretraining on *SeisNetPre* leads to the best F1 score of 95.60% while achieving the maximal accuracy and minimal error metrics on regression. The MedAE is the only metric that was minimized when an MNIST pretraining is applied. Specifically, *SeisNetPre* achieves the best F1 score of 95.60% compared to 93.25% for VGG and ResNet and 94.16% for MNIST when the fine-tuning method is used to transfer knowledge. Furthermore, two observations can be made: 1) MNIST pretraining achieves better results than VGG and ResNet, and 2) classification results are relatively high for all three networks compared to *SeisNetPre* but the regression results are worse. The first observation can be explained by the fact that MNIST training data is the most similar to seismic data as illustrated by Fig. 8. The second observation leads to the following conclusion: the bright spot detection is not highly sensitive to the pretraining method but its localization and volume estimation require precise feature representations that are best achieved in a domain-specific pretraining i.e. using *SeisNetPre* as a source task.

D. Transfer Method Effect

Both transfer methods were tested in this work: 1) using pretrained models as feature extractors and 2) fine-tuning

Fig. 9: Effect of thresholding on transfer with different pretrained networks. The first row correspond to *SeisNetTuned*, while the second, third and fourth correspond to VGG, ResNet and MNIST pretraining, respectively. The thresholds used are 0.01 (left), 10^{-5} (middle) and 10^{-8} (right). The actual output is shown in Fig. 1(Right).

them. Table III shows that the latter method is in general better than the former for different pretrained networks when considering the F1 score and the MSE. This is intuitive because fine-tuning the network not only optimizes the weights in the layers that are in charge of classification but also optimizes those that extract features from the input. However, the latter is prone to overfitting when not enough training data is available.

E. Thresholding Effect

We studied the effect of the chosen threshold on the performance. As Table III shows, the thresholding has insignificant effect on the performance of *SeisNet* when *SeisNetPre* pre-training equipped with a fine tuning transfer method is used. In contrast, VGG, ResNet and MNIST pretraining are highly sensitive to the thresholding as their accuracy and F1 score dropped by more than 60% when different thresholds are used. Fig. 9 further illustrates the effect of thresholding on the results when different pretrained networks are used. *SeisNet* appears to be the least threshold-sensitive network.

F. Deep Learning versus Shallow Learning

In their work, Rizk et al. [6] proposed a shallow learning approach that consisted of extracting statistical and textural features before training a shallow classifier in a supervised manner. They adopted as input sub-images of size 8×8

Network			Classification		Regression				
Pretraining	Transfer Method	C_2	Acc (%)	F1 (%)	MAE	MSE	MSLE	MedAE	PErr (%)
SeisNetPre	Feature extractor	E-01	93.25	87.36	125,864	1.6E10	30	132,713	50.34
		E-02	93.25	87.36	76,943	6.0E09	26	79,351	30.53
		E-05	93.25	87.36	3,810	1.5E07	10	3,949	1.51
		E-08	91.00	85.35	4,876	3.9E07	60	2,757	1.93
		E-10	91.00	85.35	4,876	3.9E07	60	2,757	1.93
		E-15	91.00	85.35	4,876	3.9E07	60	2,757	1.93
	Fine tuning	E-01	87.36	93.25	2,306	3.1E07	11	4,857	0.91
		E-02	87.36	93.25	1,218	6.0E06	8	2,016	0.48
		E-05	88.51	93.83	433	3.7E05	5	280	0.17
		E-08	91.95	95.60	503	1.8E06	3	814	0.19
		E-10	91.95	95.60	503	1.8E06	3	814	0.19
		E-15	91.95	95.60	503	1.8E06	3	814	0.19
Vgg	Feature extractor	E-01	87.36	93.25	13,529	5.0E08	17	22,219	5.36
		E-02	87.36	93.25	12,763	3.0E08	15	17,227	5.06
		E-05	87.36	93.25	7,950	1.8E08	14	13,546	3.15
		E-08	87.36	93.25	7,960	1.4E08	14	12,064	3.15
		E-10	87.36	93.25	7,960	1.4E08	14	12,064	3.15
	Fine tuning	E-01	87.36	93.25	279,487	5.0E09	24	68,997	110.90
		E-02	87.36	93.25	194,375	1.2E09	19	347,657	77.13
		E-05	87.36	93.25	187,366	2.1E08	15	14,273	74.35
		E-08	87.36	93.25	99,812	7.1E07	13	9,526	39.60
		E-10	87.36	93.25	99,812	7.1E07	13	9,526	39.60
		E-15	87.36	93.25	99,812	7.1E07	13	9,526	39.60
ResNet	Feature extractor	E-01	89.66	93.19	8,765	6.7E06	5	1,463	3.47
		E-02	87.36	92.81	6,541	2.2E06	4	537	2.59
		E-05	72.41	81.54	530	7.0E05	8	228	0.21
		E-08	24.14	23.26	109,873	2.0E07	41	1,225	43.60
		E-10	24.14	23.26	109,873	2.0E07	41	1,225	43.60
		E-15	24.14	23.26	109,873	2.0E07	41	1,225	43.60
	Fine tuning	E-01	87.36	93.25	24,592	6.5E08	16	24,623	9.75
		E-02	87.36	93.25	22,634	5.5E08	16	20,881	8.98
		E-05	87.36	93.25	19,759	4.2E08	15	17,805	7.84
		E-08	87.36	93.25	18,700	3.0E08	14	14,693	7.42
		E-10	87.36	93.25	18,700	3.0E08	14	14,693	7.42
		E-15	87.36	93.25	18,700	3.0E08	14	14,693	7.42
MNIST	Feature extractor	E-01	87.36	93.25	1,787	6.0E07	16	7,619	0.70
		E-02	87.36	93.25	1,981	6.0E07	10	2,005	0.78
		E-05	60.92	74.63	5,432	1.7E07	17	1,276	2.15
		E-08	31.03	34.78	5,514	2.0E07	36	1,358	2.18
		E-10	31.03	34.78	5,514	2.0E07	36	1,358	2.18
		E-15	31.03	34.78	5,514	2.0E07	36	1,358	2.18
	Fine tuning	E-01	89.66	94.19	90,086	6.7E06	25	14,631	35.74
		E-02	87.36	92.81	87,610	2.2E06	24	5,374	34.7
		E-05	72.41	81.54	85,012	7.0E05	18	2,286	33.73
		E-08	24.14	23.26	453,098	2.0E06	38	12,252	181.23
		E-10	24.14	23.26	453,098	2.0E07	38	12,252	181.23
		E-15	24.14	23.26	453,098	2.0E07	38	12,252	181.23

TABLE III: Classification and regression results for SeisNet with different pretraining methods.

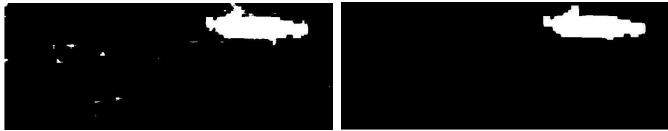


Fig. 10: Left: \hat{y} matrix. Right: \hat{y} matrix after morphological opening with a disk of radius 10 pixels as a structuring element.

Classification		Regression					
Acc (%)	F1 (%)	MAE	MSE	MSLE	MedAE	PErr (%)	
Pre	91.95	95.60	503	1.8E6	3.0	814	0.19
Post	91.95	95.60	143	9.3E4	2.7	120	0.04

TABLE IV: Performance before and after a morphological opening for the best *SeisNet* model

and a simple binary output. For a fair comparison with their approach, we divided the testing dataset into 8×8 sub-images that will be instead used to compare *SeisNet* performance to that of [6]. We divided our label matrix \hat{y} , in a similar manner, into 8×8 sub-matrices. We adopted the same approach as in [6], i.e. a sub-image is classified as 1 if the majority of its pixels are labeled as bright spot, in the labels sub-matrix, and 0 otherwise. Testing revealed an F1 score of 95.6% for *SeisNet* compared to 85.4% reported in [6] and obtained by a non-iteratively trained ANN on feature vectors of dimension 48.

G. Morphological Opening

Fig. 9 showed that the threshold value may result in erroneous bright spot predictions that are also caused by the inherent noisiness of seismic data. Therefore, we investigated the effect of applying morphological opening on the volume estimation. One can clearly notice that some independent pixels might be labeled as 1 without constituting a bright spot with a significant volume due to the noisy nature of seismic data. A morphological opening would remedy such misclassifications as Fig. 10 depicts by further processing the matrix \hat{y} , that is mapped to the volume of the bright spot, with a disk of radius 10 as a structuring element. Table IV illustrates that the morphological opening did not affect the classification performance but improved the regression metrics by a factor of around 4 for the MAE and of 20 for the MSE error.

H. *SeisNet* for Other Seismic Applications

SeisNet was empirically shown to perform well on bright spot detection. Being trained on a large dataset of diverse seismic images, *SeisNet* has the potential to outperform other pretrained networks on different seismic applications. In this section, we show how *SeisNet* outperforms state-of-the-art work on two different seismic applications: fault segmentation and facies classification. Readers who are interested in the problem statement and the description of the data used in both applications are referred to [32] and [47] respectively. In

[32], Wu et al. cast the fault dection as an image segmentation problem and used a simplified version of U-Net [48] to train a CNN that achieved a 95% pixel accuracy. Using *SeisNet* as a pretrained network, our method outperformed that of [32] by achieving a similar accuracy of 94% when transfer learning is not used and a higher accuracy of 99.98% when both transfer learning methods described in this paper are used.

On the other hand, Alaudah et al. [47] used deconvolution networks to solve the facies classification problem. Table V reports the evaluation metrics used in [47] to test the performance of the approach. We first report the metrics on the best model as done in [47]. We then report the performance of *SeisNet* when the transfer learning is not used, i.e. when *SeisNet* is trained from scratch, and its performance when transfer learning is applied using the feature extraction (fixing weights) and the fine-tuning method. The best results on the pixel accuracy and the mean class accuracy (95.7% and 97.2%) are attained when *SeisNet* is used while fine-tuning the weights as opposed to 90.5% and 81.7% as reported by Alaudah et al. in [47]. The frequency-weighted intersection over union achieves better results when the feature extractor method is used (84% versus 83.2%).

VI. CONCLUSION

In this work, a novel deep transfer learning formulation for bright spot detection from seismic data allowed the detection of a bright spot and estimation of its volume with high accuracy. *SeisNet*, a deep CNN with a “butterfly” architecture, outperformed other approaches in the literature by achieving a F1 score of 95.6% on the classification task and estimated the volume accurately with no more than 0.04% of the pixels classified incorrectly on the volume prediction task. We employed data augmentation and inductive transfer learning to overcome the limited data problem that hinders the effectiveness of deep learning algorithms. Data augmentation was shown to enhance the F1 score by approximately 50% and transfer learning improved it by almost 10%. Studying the effect of the latter technique, we noted that better performance was achieved when fine tuning was used to transfer knowledge from the source domain to the target domain.

Moreover, a domain-specific source task, instead of general domain tasks, improved the performance of deep learning on the classification and regression tasks. Specifically, *SeisNet* used as a pretrained network improved the performance of two seismic applications: fault segmentation and facies classification by 5.2% and 5.7% respectively. Therefore, our contributed *SeisNet* model should be used as a domain-specific pretrained network to jump-start the training of deep neural networks on different seismic applications. Future work will investigate the performance of our approach and model to other seismic data analysis problems such as earthquake prediction based on waveform attributes, the effectiveness of other types of deep neural networks on seismic data analysis within our framework and the performance on larger datasets with varying degrees of noise.

Model	PA (%)	Class Accuracy (%)						MCA (%)	FWIU (%)
		Zechstein	Scruff	Rijnland	Lower N.S.	Middle N.S.	Upper N.S.		
[47]	90.5	60.2	67.4	77.2	94.1	93.8	97.4	81.7	83.2
SeisNet + No transfer	81.7	58.3	65.0	75.2	94.8	94.9	99	79.1	79.9
SeisNet + Feature extractor	95.5	69.1	69.7	82.1	98.3	97.6	93	95.8	84.6
SeisNet + Fine tuning	95.7	69.3	69.2	82.3	99.8	97.6	96	97.2	84.5

TABLE V: Comparison with [47] where PA, MCA and FWIU mean pixel accuracy, mean class accuracy and frequency-weighted intersection over union respectively.

ACKNOWLEDGMENT

This work was funded by the University Research Board at the American University of Beirut and the Munib Masri Institute.

REFERENCES

- [1] O. A. El Souki, G. A. Saad *et al.*, “A stochastic approach for optimal sequencing of appraisal wells,” *SPE Reservoir Evaluation & Engineering*, vol. 20, no. 02, pp. 334–341, 2017.
- [2] E. Darko, *Short guide summarising the oil and gas industry lifecycle for a non-technical audience*. London: Overseas Development Institute, 2014.
- [3] Ö. Yilmaz, *Seismic data analysis*. Soc. of exploration geophysicists Tulsa, 2001, vol. 1.
- [4] J. F. Claerbout, “Digital filters and applications to seismic detection and discrimination,” Ph.D. dissertation, Massachusetts Institute of Technology, 1963.
- [5] L. C. Duval, V. Bui-Tran, T. Q. Nguyen, and T. D. Tran, “Genlot optimization techniques for seismic data compression,” in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4. IEEE, 2000, pp. 2111–2114.
- [6] Y. Rizk, H. Partamian, and M. Awad, “Toward real-time seismic feature analysis for bright spot detection: a distributed approach,” *IEEE J. Sel. Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 1, pp. 322–331, 2018.
- [7] K. G. Martin, M. W. Totten, and A. Raef, “Characterization of a reservoir ooid shoal complex and artificial neural networks application in lithofacies prediction: Mississippian st. louis formation, lakin fields, western kansas,” *J. Petroleum Sci. Eng.*, vol. 150, pp. 1–12, 2017.
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [9] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Le-Cun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv preprint arXiv:1312.6229*, 2013.
- [10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [11] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” in *Advances in neural information processing systems*, 2012, pp. 2843–2851.
- [12] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt *et al.*, “From captions to visual concepts and back,” in *Proc. IEEE Conf. computer vision and pattern recognition*, 2015, pp. 1473–1482.
- [13] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *Proc. IEEE Conf. computer vision and pattern recognition*, 2015, pp. 3128–3137.
- [14] T. Perol, M. Gharbi, and M. Denolle, “Convolutional neural network for earthquake detection and location,” *Sci. Adv.*, vol. 4, no. 2, p. e1700578, 2018.
- [15] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, “When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 56, no. 5, pp. 2811–2821, 2018.
- [16] F. Yang and J. Ma, “Deep-learning inversion: a next generation seismic velocity-model building method,” *Geophysics*, vol. 84, no. 4, pp. 1–133, 2019.
- [17] X. Wu, Y. Shi, S. Fomel, L. Liang, Q. Zhang, and A. Z. Yusifov, “Faultnet3d: Predicting fault probabilities, strikes, and dips with a single convolutional neural network,” *IEEE Transactions on Geoscience and Remote Sensing*, 2019.
- [18] H. Di, D. Gao, and G. AlRegib, “Developing a seismic texture analysis neural network for machine-aided seismic pattern recognition and classification,” *Geophysical Journal International*, vol. 218, no. 2, pp. 1262–1275, 2019.
- [19] X. Wang, B. Wen, and J. Ma, “Denoising with weak signal preservation by group-sparsity transform learning,” *Geophysics*, vol. 84, no. 6, pp. 1–90, 2019.
- [20] H. Di, Z. Wang, and G. Alregib, “Deep convolutional neural networks for seismic salt-body delineation,” *AAPG Annual Convention and Exhibition*, pp. 1–12, 2018.
- [21] Y. Zeng, K. Jiang, and J. Chen, “Automatic seismic salt interpretation with deep convolutional neural networks,” *arxiv:1812.01101v1*, pp. 1–11, 2018.
- [22] L. Mou, P. Ghamisi, and X. X. Zhu, “Deep recurrent neural networks for hyperspectral image classification,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.
- [23] L. Huang, X. Dong, and T. E. Clee, “A scalable deep learning platform for identifying geologic features from seismic attributes,” *The Leading Edge*, vol. 36, no. 3, pp. 249–256, 2017.
- [24] M. Titos, A. Bueno, L. Garcia, and C. Benitez, “A deep neural networks approach to automatic recognition systems for volcano-seismic events,” vol. 11, no. 5, pp. 1533–1544, 2018.
- [25] M. Araya-Polo, T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, “Automated fault detection without seismic processing,” *The Leading Edge*, vol. 36, no. 3, pp. 208–214, 2017.
- [26] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke, “Deep-learning tomography,” *The Leading Edge*, vol. 37, no. 1, pp. 58–66, 2018.
- [27] S. Kanarachos, S.-R. G. Christopoulos, A. Chroneos, and M. E. Fitzpatrick, “Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and hilbert transform,” *Expert Systems with Applications*, vol. 85, pp. 292–304, 2017.
- [28] Y. Shi, X. Wu, and S. Fomel, “Saltseg: Automatic 3d salt segmentation using a deep convolutional neural network,” *Interpretation*, vol. 7, no. 3, pp. SE113–SE122, 2019.
- [29] M. Fehler and P. J. Keliher, *SEAM phase 1: Challenges of subsalt imaging in tertiary basins, with emphasis on deepwater Gulf of Mexico*. Society of Exploration Geophysicists, 2011.
- [30] N. Pham, S. Fomel, and D. Dunlap, “Automatic channel detection using deep learning,” in *SEG Technical Program Expanded Abstracts 2018*. Society of Exploration Geophysicists, 2018, pp. 2026–2030.
- [31] T. Zhao, “Seismic facies classification using different deep convolutional neural networks,” in *SEG Technical Program Expanded Abstracts 2018*. Society of Exploration Geophysicists, 2018, pp. 2046–2050.
- [32] X. Wu, L. Liang, Y. Shi, and S. Fomel, “Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation,” *Geophysics*, vol. 84, no. 3, pp. IM35–IM45, 2019.
- [33] F. Huot, B. Biondi, and G. Beroza, “Jump-starting neural network training for seismic problems,” in *SEG Technical Program Expanded Abstracts*. Society of Exploration Geophysicists, 2018, pp. 2191–2195.
- [34] B. Peters, J. Granek, and E. Haber, “Multi-resolution neural networks for tracking seismic horizons from few training images,” *arXiv preprint arXiv:1812.11092*, 2018.
- [35] S. Chopra and V. Alexeev, “Applications of texture attribute analysis to 3D seismic data,” *The Leading Edge*, vol. 25, no. 8, pp. 934–940, 2006.
- [36] M. Noris and A. Falchney, “Segy rev1 data exchange, format 1,” *Tulsa, OK: Society of Exploration Geophysicists*, 2002.
- [37] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, “Understanding data augmentation for classification: when to warp?” in *Int.*

- Conf. digital image computing: techniques and applications.* IEEE, 2016, pp. 1–6.
- [38] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, and Z. Jin, “Improved relation classification by deep recurrent neural networks with data augmentation,” *arXiv preprint arXiv:1601.03651*, 2016.
- [39] C. N. Vasconcelos and B. N. Vasconcelos, “Increasing deep learning melanoma classification by classical and expert knowledge based image transforms,” *CoRR, abs/1702.07025*, vol. 1, 2017.
- [40] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [41] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [42] G. E. Hinton and R. S. Zemel, “Autoencoders, minimum description length and helmholtz free energy,” in *Advances in neural information processing systems*, 1994, pp. 3–10.
- [43] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in neural information processing systems*, 2014, pp. 3320–3328.
- [44] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Proc. IEEE Conf. computer vision and pattern recognition workshops*, 2014, pp. 806–813.
- [45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. computer vision and pattern recognition*, 2016, pp. 770–778.
- [46] Y. LeCun, C. Cortes, and C. Burges, “Mnist handwritten digit database,” *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, p. 18, 2010.
- [47] Y. Alaudah, P. Michalowicz, M. Alfarraj, and G. AlRegib, “A machine learning benchmark for facies classification,” *Interpretation*, vol. 7, no. 3, pp. 1–51, 2019.
- [48] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.



Mariette Awad (M’08) obtained her PhD in Electrical Engineering from the University of Vermont (2007). Her current research focuses on HMI, artificial intelligence, machine learning, data analytics and Internet of Things. Dr. Awad has received 21 grants to support her research including 2 multidisciplinary multi-million dollar grants from the Qatar National Research Fund (QNRF) and Intel. Her work culminated in a book, Efficient Machine Learning, in 2015 as well as more than 70 conference, book chapter, and journal publications. Awad has been an invited speaker and reviewer at a number of international conferences and journals. Prior to her academic position, she was with IBM System and Technology group in Vermont for six years as a wireless product engineer where her technical leadership and innovative spirit has earned her management recognition twice, two business awards, and 10 patents.



Julia El Zini is a PhD student enrolled in the electrical and computer engineering department at the American University of Beirut (AUB). She has received her BS and MS in computer science from AUB, Lebanon, in 2015 and 2017, respectively. Her research interests include distributed optimization, parallel computing, reinforcement learning, multi-task and transfer learning, and scalable machine learning applications.



Yara Rizk obtained her PhD in Electrical and Computer Engineering from the American University of Beirut (AUB) in 2018. Prior, she received her BE in Computer and Communication Engineering from AUB, Lebanon, in 2012. Her research interests span robotics, multi-agent systems, machine learning, classification, clustering, and artificial intelligence. Rizk has attended a technical internship (2013-2014) at Intel in Hillsboro, Oregon, USA and is an active researcher multiple peer-reviewed publications.