A Deep Neural Network Framework for Seismic Image Classification & Analysis

MSc Research Project
Data Analytics

Anusha Gorur Chandrashekar
Student ID: x18195059

School of Computing
National College of Ireland

Supervisor: Dr.Pramod Pathak, Dr.Paul Stynes

National College of Ireland
MSc Project Submission Sheet
School of Computing

Student Name:	Anusha Gorur Chandrashekar
Student Id:	X18195059
Programme:	MSc Data Analytics
Year:	2019-2020
Module:	MSc Research Project
Supervisor:	Dr.Pramod Pathak, Dr.Paul Stynes
Submission Due Date:	17/08/2020
Project Title:	A Deep Neural Network Framework for Seismic Image Classification & Analysis
Word Count:	8416
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	**Anusha Gorur Chandrashekar**
Date:	**17/08/2020**

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Deep Neural Network Framework for Seismic Image Classification & Analysis

Anusha Gorur Chandrashekar
x18195059

Abstract

Seismic imaging is an indispensable tool in oil and natural gas (O&G) exploration as it helps to delineate the earth's subsurface structures and monitoring through the geometry of sources-receivers coordinates. In the contemporary era, it demands large volumes of seismic data be analyzed and interpreted under standard procedures. Therefore, computational framework systems can stimulate the expert in the classification of subsurface structures/facies to accelerate the analysis process which is paramount to the growth of the industry. The rise in popularity of deep learning motivated scientists to extend those methods to 3D seismic data. While this technique manifested positive returns, the complexity of finding a good starting point for optimizing the parameters of the model is a conventional problem in deep learning systems. Poor or random initialization may lead the network to longer training sessions, vanishing gradients due to backpropagating till initial layers and failing to find the solution. To address this issue, the use of transfer learning with state-of-art deep neural network models are utilized to set a good initialization point to the parameters of the neural network models. The data used is F3 Dutch seismic cube data widely available as open-source data. In seismic data, different facies can be identified by differences in the signatures of the amplitudes, which allows us to label different facies, to assist the process nine facies labels of the data is available from Project MalenoV repository. To achieve this, leveraging the transfer learning with the use of pre-trained models VGG-16, ResNet and Efficient B7 architectures to analyze and interpret the seismic facies. Each model is optimized and evaluated using classification and regression metrics such as Precision, Mae, Categorical accuracy. As a result, VGG-16 and Efficient B7 outperformed ResNet with accuracy 97.4% and 99.1%, MAE 0.085, and 0.008 respectively.

***Keywords*:** Seismic Interpretation, Transfer Learning, Convolutional Neural Networks, Oil, and Gas Exploration.

1 Introduction

1.1 An overview of Seismic Imaging Classification

Seismic facies classification is reliant on the knowledge and expertise of the interpreters. However, deep learning cannot suffice the specialized expertise, in analyzing seismic data although it can significantly simplify the process of interpreting facies and mapping geological regions. This confluence of man and machine can significantly enhance speed, quality, knowledge of geological mapping which appears to be the rational solution to improve seismic imaging classification. Conversely, seismic data interpretation is a prolonged and labor-intensive task. Besides this geologists and geophysicists must dispense with an enormous volume of data continually. In this existent scenario, it inspires to propose an auxiliary computational system to help the interpreters, in reducing the time and to increase the accuracy of the interpretation task. As shown in Figure 1, the major aspect is the label-dependent feature extraction. The motive of this aspect is twofold, on one hand, to simplify the data samples for training to reduce the volume of data, to alleviate the complexity on the computing resources, at the same time, it enables the training to emphasis on the definite feature within the seismic data that are relevant to this experiment, which can give rise to the improvement of the accuracy and sharp prediction of fault labels [21].
The actual deep neural network process begins after the feature extraction and the use of advanced transfer learning algorithms and custom layers look to enhance each step of the workflow through a profound knowledge of the data.

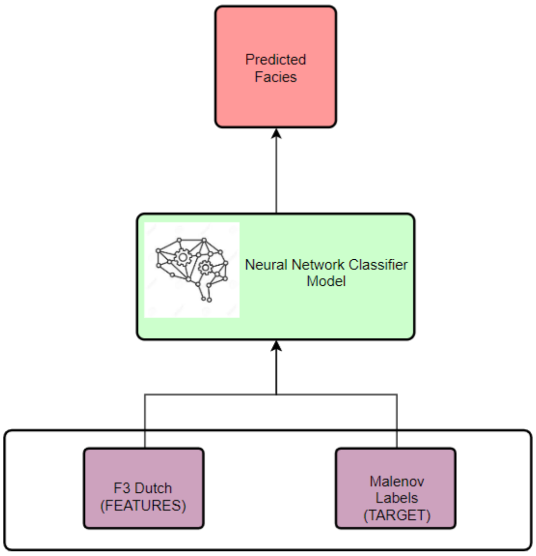

Figure 1- Seismic classification approach

With the growth of computational powers and deep neural networks which is capable of solving the complexity of seismic data interpretation yielding high-quality results and assisting the geologists and geophysicists in interpreting geological structures in oil and gas exploration process. This research has aimed to extend the work utilizing state-of-the-art Convolutional Neural Network architectures such as VGGNet, ResNet, and Efficient B7 to fully exploit its potential on the seismic facies classification problem.

1.2 Background and Motivation

Petroleum exploration is the method by which the oil and gas resources are discovered. Seismic surveys and studies play a vital role in petroleum exploration because of their proven track record of success and high efficacy in identifying the location of oil and gas deposits. Modern seismic surveys have become the primary tool of energy exploration companies both onshore and offshore. It requires a large amount of data, manpower, computer clusters. Seismic data acquisition requires an energy source to generate the seismic wiggle trace/amplitude and sensors at the receiving end. The collected raw seismic signatures are processed to produce an ordered series of images illustrating the earth's subsurface that lie beneath the rock layers. The fashion of these reflected signals gives geophysicists an idea of the type of structures. Below figure 2 illustrates the offshore seismic acquisition survey [1].

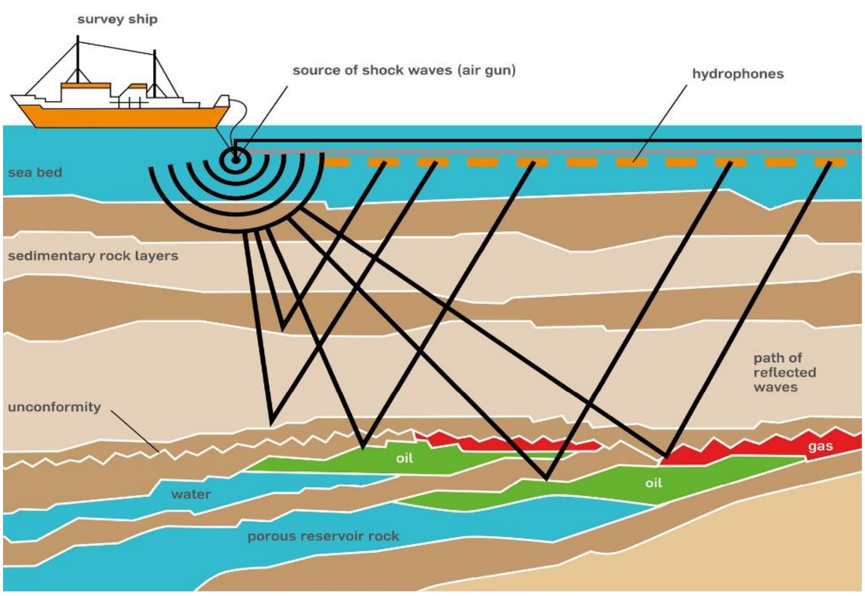

Figure 2- Offshore seismic acquisition survey.

Identifying, characterizing, and tracking oil and gas reservoirs is a critical step of oil and gas production as companies invest billions of dollars for almost a decade. Erroneous interpretation of the data may cause inaccurate decisions costing billions of dollars in losses for energy firms. The product of the seismic survey gives the noisy seismic reflection 2D/3D data which is used to allocate the oil, gas, and minerals. Recently, different technologies such as Machine Learning, Artificial Neural Networks (ANN) have been investigated in processing the noisy data and extract useful features with a reduced error rate [1]. Convolutional Neural Network a variant of ANN has also been successfully applied to the seismic data analysis.

However, there are many challenges that make the seismic facies classification more complex, firstly typical image classification problem aims at distinguishing the images, whereas seismic facies classification tends to distinguish various geological structures in the same image. Secondly, the availability of the seismic training data is sparser compared to other image classification problems. Thirdly, in the seismic image data, all the features are rarely explicitly defined due to their complex patterns of reflectors. These addressed challenges highly motivate the need for the research study, hence the CNN models with transfer learning are attempted to automate the seismic facies interpretation model.

1.3 Research Question

"To what extent Deep Machine Learning approaches help in identifying the geophysical structures (such as faults, channels,) in 3D seismic reflection data using multiple seismic attributes (such as inline, crossline, amplitude, time slice) ?"

This research aims to address the above research question by performing seismic facies analysis using deep neural network transfer learning models and by merging multiple attributes like inline, crossline, amplitude, time slice. This analysis of seismic data might help in acquiring useful insight that will benefit Geophysicists, Geologists, Oil, and Gas industries. Since the output to be predicted in this study is the classification of facies in the 3D seismic reflection data, neural network architectures (such as VGGNet, ResNet v1, v2) are used in the implementation of this seismic facies classification model.

1.4 Research Objectives

Following research objectives are pursued in order to answer the above research question:

- Pre-processing of seismic data that deals with image data slicing, correction of noise, signal-enhancement, trace editing.
- Identifying factors impacting on the classification of facies by exploratory analysis and creating a GUI to speed up the visualization.
- For the Fine-tuning process, optimization method such as Stochastic Gradient descent with momentum is used.
- Implementation of Deep Neural Network models on the feature engineered seismic data using VGGNet, Residual Neural Network (ResNet).
- To improve Regularization and avoid overfitting of the model, using dropout and batch normalization techniques to the network.

This paper is organized as follows: Section 2 elaborates the related work carried out in this area, Section 3 describes the methodology followed in this research work, Section 4 presents the experimental setup/design flow, Section 5 demonstrates the neural network model implementations, Section 6 shows the evaluation of results combined from the individual stages of the flow. Finally, section 7 presents the conclusions of the study.

2 Related work

2.1 Deep learning for seismic data analysis

Deep learning has been applied to solve many real-world problems in the same way as the human brain would from natural language to scene and speech recognition in various domains like healthcare, transportation, geoscience, and remote sensing, etc. Converging on seismic data analysis, machine learning techniques have been attempted to solve various geological problems such as seismic velocity model building, geological structure recognition, seismic pattern classification, and fault probabilities. Convolutional neural network (ConvNets) has been a hot topic in the seismic image analysis [3] adopted U-net architecture and ResNets with best network adjustments such as Exponential Linear Units (ELU) activation function, Lovsaz softmax loss function with adam optimizer to salt body interpretation on the seismic image. Overall the performance indicates improved accuracy than other approaches mentioned in their literature review. The work proposed by Huang *et al* in[4] efficiently combined traditional machine learning algorithms like support vector machine (SVM), logistic regression, and CNN with seismic attributes to attain scalable performance in identifying the faults providing the distributed-memory infrastructure with customized seismic analytics software development toolkit (SDK). A novel approach challenging the multi-step seismic model building problem was presented by Polo et al. [5] to eliminate the expensive physical modelling by adopting a data-driven approach and a basic neural network to automate fault detection. In the preceding model, the Wasserstein loss function is used to tackle the spatial layout dependency problem on the outputs; experiments witnessed promising results achieving an area under the ROC curve (AUC) value of 91% on the synthetic data [21]. Jetley et al. [6] proposed progressive U-net architecture with a soft attention mechanism and a dilated convolutional network is applied to the seismic data. A fair comparison is made among 4 different models such as dilated U-Net, dilated U-Net with attention, U-Net, U-Net with attention. With upsampling features from 456 x 944 to 462 x 951, for the dilated U-Net with attention provided clean seismic facies classification showing IOU 0.883 with only fewer parameters compared to other models. From the perspective of seismic fault classification, automatic facies detection has been the main research focus on the progress of computer graphics and image processing. A 1-layer CNN classifier architecture was demonstrated in [7] to interpret the facies using volumetric image processing with an input image of 32 x 32 and 2 x 2 max-pooling is adopted to avoid overfitting by reducing the dimension of the output features. The results cast that a simple CNN was able to classify the faults by learning the target seismic features from the original post-stack seismic amplitude. [8] also uses deep convolutional neural network-based auto-encoder architecture with skip connections to extract only useful information from the augmented training data and validated on the synthetic data, yielding high accuracy and great generalization over different 3D seismic data. Waldeland et al. [9] also demonstrated CNN on the raw seismic data and compared the model results after fine-tuning with best parameter settings, like reducing cost function, applying batch norm before dropout, and SGD optimizer.
The above-mentioned references are the deep neural network model built from scratch. However, there are few circumstances, where the availability of the data is a concern or data labelling can be expensive. Some works in the below present transfer learning approaches that do not need a significant amount of the data.

2.2 Survey on Transfer Learning

Several research has implemented transfer learning to analyze seismic facies. This deep learning approach allows the transfer of the previous knowledge acquired when solving one problem and is reprocessed with a faster and more reliable solution for the second problem. The idea behind CNN's transfer learning is to freeze the previous convolutional layers and use only the last layers for the prediction. This segment summarizes and provides specifics of previous studies that point to the general functioning and the implications of the transfer of learning applied to seismic data. The above sections discussed employed deep deep neural network models from scratch. However, in certain scenarios, data availability is a concern. Data labelling is an affluent task and time-consuming. Many of the approaches suggested in the literature do not need more data. Chevitarese et al. [10] conducted experiments on two alternatives for the advantage of transfer learning. Scenario 1: The data available is sparse, constrained, and inadequate to train CNN; scenario: training the model from baseline can be challenging and parameter initialization is complex. A poor starting point could result in longer training session time and impotence to reach a solution. Transfer learning is established to solve these complexities in order to set a good initialization to the parameters of the model and define the values of parameters acquired from the previous knowledge. The result yielded in this work, shows that transfer learning improved the model accuracy and enabled to train the model which had a random initialization problem.

The inductive transfer learning and data augmentation methods are adopted in [2] to overcome the limited data issue. This study implemented “Seisnet” a novel approach that casts synthetic data generated from the limited dataset, a source task autoencoders to compress the input data into shorter code and uncompress to the original data from the shorter code and then this architecture is tested against VGGNet, ResNet transfer learning methods. The combination matrix formed by data augmentation of labelled data which is passed into Seisnet and unlabeled data passed into the autoencoder is deployed to predict the faults, channels, and bright spots. To summarize, the issue of data limit is resolved by using data augmentation and transfer learning, achieving 95.6% accuracy in the classification, and transfer learning enhance by10% gain, surpassing previous research approaches in the literature they mentioned.

To make a fair comparison of the transfer learning models cast in this work is improved by employing several fine-tuning techniques in two of the experiments demonstrated in sections 6.1 and 6.2.

2.3 Model Scaling

There are several ways to scale the ConvNet by using the common principle methods such as increasing the number of layers (depth), increasing the input images (resolution). The work proposed in [11] focuses on scale down (ResNet-18) and scale-up of ResNet (ResNet-200) by adjusting the layers i.e. depth. In the first comparison of ResNet 18 and 34, Resnet-34 exhibits better results in achieving lower training error reducing to 3.5%, and generalizable to the validation set. This comparison addresses the degradation problem well and manages to obtain greater accuracy as the number of layers increased. Later experiments were constructed with 101 and 152 layers remarkably increasing the layers, which is still lesser complex than Vggnets, witnessed in greater accuracy and lower training error. This single model Resnet 152 outperformed all the other ensemble results. Though the problem of very deep neural networks is that features get diminished, slow training, also double the layers to improve a fraction of accuracy which is expensive. To tackle these issues [12] has proposed a novel approach where increasing the width and decreasing the depth of the residual network. The settings of depth and width are carefully chosen for instance WRN 28-10, WRN 40-10 (Wide Residual Network) have 3.6 and 5 times more parameters than ResNet having 1001 layers and both WRN models outperformed by a significant margin. Similarly, Howard et al. [13] proposed a width multiplier convolutional neural network architecture called MobileNets trading off a reasonable dropping of accuracy until the architecture shrinks to reduce the network size and latency to build a fast and smaller network. Following the same principle [14] focused on model scaling such as width, depth, and resolution scaling on convolutional neural networks. The conclusion is drawn in experimenting ResNet model scaling on scaling dimensions (width, depth, resolution) that includes as increasing depth (d) reached saturation while ResNet 1001 and ResNet 101 gained the same accuracy, increased width (w) resulted in capturing fine-grained features and much easier to train, whereas the accuracy rapidly saturates as the network grows wider than larger, higher resolution (r) exposes much detail to capture, but a decreased accuracy. A valuable finding is drawn from the above conclusion, that is the dimension (w,d,r) is not independent, as there is a need to increase in depth of the network to capture many details when the resolution of the input is increased. Based on this, the new architecture named “EfficientNet” is targeted on computational efficiency. The EfficientNet is compared with various models such as ResNets, MobileNets, and exhibited greater accuracy with only a few parameters. Hence the balanced maintenance of dimensions “compound scaling” obtained an EfficientNet model. Using this idea to scale up the model, increasing the dimensions in the constant ratio is the best way to achieve the model accuracy, and it is witnessed in the experiment section 6.3 yielding the best performance on seismic data than previous ConvNets.

2.4 Performance comparison of recently used CNNs

In recent years, deep learning has brought the breakthrough in seismic analysis. In that U-net architecture developed by Olaf Ronneberger et al [15] for biomedical image segmentation problems, the same idea is utilized in [16] to tackle a problem of classification of seismic facies and introduced the novel idea using the U-net model inspired by ResNet and DenseNet architecture. The network architecture is composed of 3 blocks and each block is designed as the complex network with 5 convolution layers. During the training process, two levels of data augmentation were incorporated, of which the first level was to double the training set by horizontal image flipping and coordinating masks. The second stage has focused on the random image transformation featuring intensity reform and band scaling. Wu et al. [17] suggested a CNN Classifier model detects the fault on the synthetic seismic images, relying on binary classification with 0's being no faults and 1's being a fault. The synthetic data formed with 200 samples appeared biased and inconsistent class distribution, hence using "class balanced binary cross-entropy loss function" to avoid the model's biased prediction and optimizing the parameters to achieve the pixel accuracy of 95%. However, there are certain complexities in applying CNNs, to tackle this issue intersection of Bayesian probability theory and deep learning is implemented in [18] study. Besides, using regularization techniques, SGD, Monte Carlo dropout technique to estimate the uncertainty of prediction applied on both test and train set and to pass the data on loop, collecting all the predictions. in conclusion, showing that the Bayesian neural network is faster and more efficient than the approaches traditionally considered. Autoencoders have been implemented in the work carried out by [19] in comparison to the patch-based model. The patch-based model gathers data on training and testing, which typically shows the most influential image samples showing the highest confidence features. Thus autoencoders attempt to classify the target facies. Qian et al. [20] developed a multi-layered, prestack data-based, convolutional encoder. In each layer, higher-level feature maps are created by transforming lower-level features with convolution filters that reduce the error function. By attempting to learn reliable features from the prestack data, remarkable classification has been observed compared with other ensemble methods.
From the above-related work, it is significant that transfer learning and CNNs has proven to be an efficient combination to train the model and analyze the complex seismic image dataset with multi-attribute analysis encompassing the use of different data attributes to predict various geological features.

3 Methodology

This research study methodology resembles the open standard process model called the cross-industry process for data-mining (CRISP-DM). The figure below represents the structured approach in designing the methodology of the seismic facies classification covering stages namely: business understanding, data acquisition, pre-processing, modelling, evaluation, and predicted result. The result generated could be used for the effective classification of facies.

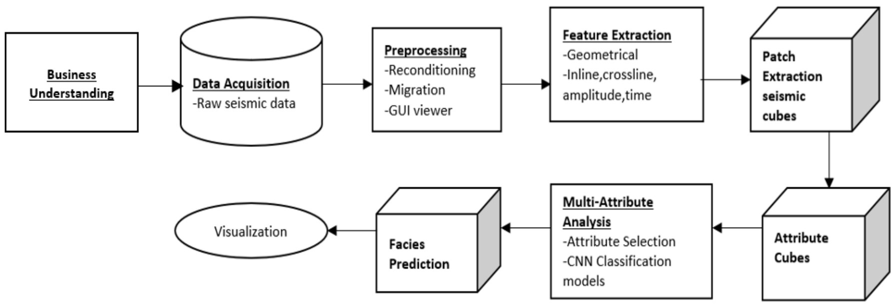

Figure 3- Seismic facies classification- Design Methodology

3.1 Business Understanding

Oil and gas industries are continuously dealing with various industry-specific challenges, like lack of coverage in complex operational processes, performance enhancement issues, life cycle management of equipment, the difficulty of logistics, and compliance with environmental regulations. Through big data analytics companies transform large data sets into oil and gas exploration decisions, reduced computational costs, enhanced equipment longevity, and lower environmental impact. Exploration processes are under great pressure, from enhancing performance at a lower cost to constant data avalanche from futuristic generations of sensors and modern acquisition systems [21]. The industries use a seismic technique to survey the field and determine whether the area in question contains reserves of oil and gas. Most of the key steps in these business processes are dependent on domain experts, their time is limited, but the volume of data that needs to be analyzed in depth is increasing. Moreover, the difficulty of some of the areas of research demands particular attention. The problem is defined as a data explosion, which is becoming more complicated than ever before. Hence, the advanced data-oriented algorithms aim to enhance each stage of the workflow by a deeper understanding of the data, from capturing the relevant information to better comprehension and integrate new tools to empower geoscientists.

3.2 Data Acquisition

The seismic data used in this research study is Netherlands offshore F3 seismic data extracted from the location of North Sea, Netherlands offshore which is available at the open-source seismic repository DGB earth sciences[1]. The dataset consists of 3D images of 651 inlines and 951 crosslines with the time range of 1848ms. This seismic data is interpreted by geoscientists and provided 9 different labels depending on the significant texture of the seismic images in the project called Malenov and that labeled data is also available in the public geoscience repository. The analysis of the seismic facies leads to the identification of various geological features such as faults, channels interpreting oil, gas reservoir, and deposition regions. Below is the table which describes various facies on the rationale of the amplitude and continuity of waveforms recorded seismic images.
The nine horizons/facies were interpreted as follows-

Table 1- Facies representation with labels

Labels	Facies	Color code
0	else	turquoise
1	Lower coherency	brown
2	Steep dipping reflector	gray
3	Low amplitude dipping reflector	green
4	Continuous high amplitude	blue
5	grizzly	orange
6	Low amplitude	yellow
7	High amplitude	magenta
8	Salt intrusions	gray

[1] https://terranubis.com/datainfo/Netherlands-Offshore-F3-Block-Complete.

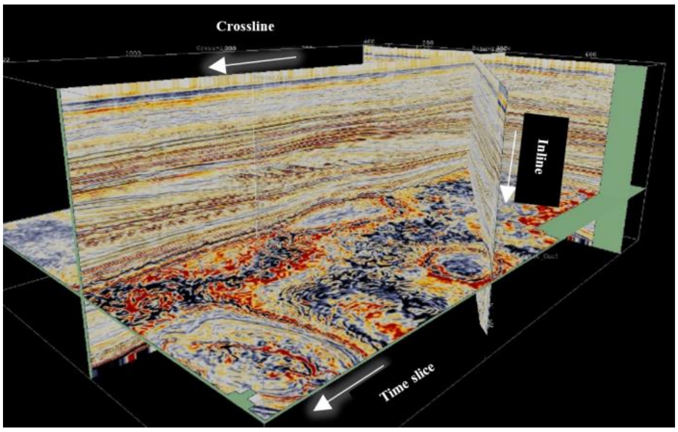

Figure 4- 3D seismic volume interpretation of inline, crossline and time slice

3.3 Data Pre-processing and Analysis

Before feeding the data into the neural network, the data must be preprocessed. In this study experiment, the following preprocessing methods are used:

Seismic Image Data Slicing- The F3 Netherlands seismic data is a formulated cube data, which is a 3D array of seismic amplitudes, processed in different dimensions and this is fed into the memory using the segyio library in python. The seismic data is a matrix where index into a sequence is done to select the sub-elements or specific slices from the whole seismic cube. Slicing and indexing is a straightforward task using the NumPy numerical library for crunching numbers. Here is an example used in the experiment to slice images from the cube:

-timeslice=data[:,:,339] #The 339th image slice from the third dimension
-inline=data[500,:,:] # The 500th image slice from the first dimension.
-crossline=data[:,62,:] # The 62nd image slice from the second dimension.

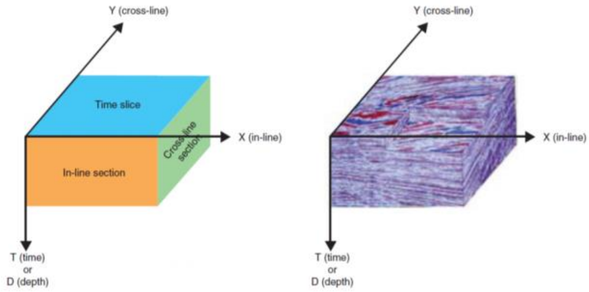

Figure 5- Representation of time, inline, crossline of the 3D seismic volume

Data selection- The selection of the image slices is the crucial task to be made. Choosing the image slices closest to the manual marking is the key trick to acquire. Detailed image analysis is the appropriate way to observe the images, hence the Graphical User Interface (GUI) is employed in the experiment to ease the selection of the image slices. The standard display of image slices with the displaying of variance in seismic attributes helped in understanding the nature of the images based on the amplitude signatures from the seismic images. Having looked through all the slices, inline crossline ranging beyond 300th slice shows clearer enhanced attributes, and few distorted images did not present useful information. Below are the image slices viewed using GUI.

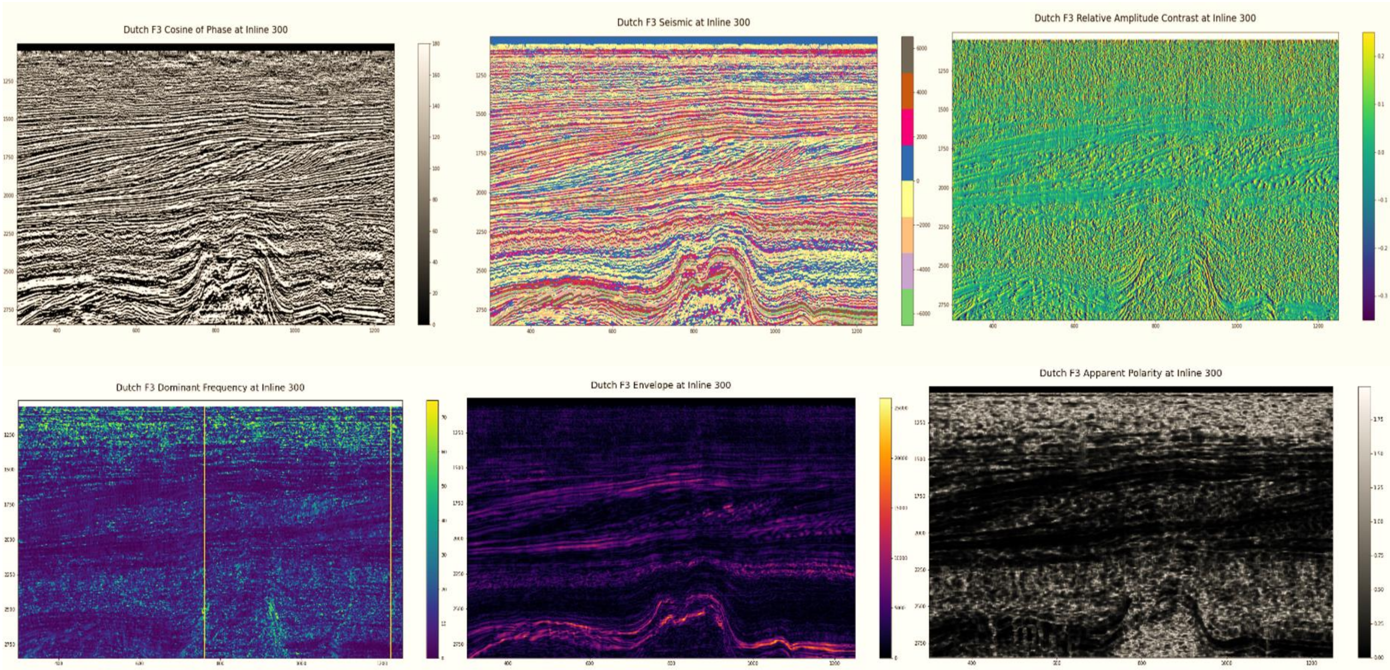

Figure 6- GUI viewer for data selection

Patch extraction- Before training a neural network, 64 x 64 2D patches are created in the crossline and inline direction. The patch_extractor_2d function is used in extracting the patches from an image stored as a 2-dimensional array, or 3-dimensional array including color information. The advantage of this approach is that generating thousands of training patches from one annotated image as seismic data are hard to collect and annotate and it is the complex task to provide million of images with annotation and train end-to-end system. The process in this patch extraction is simple as follows, firstly, annotation of publicly available seismic data, then based on these annotations extracting 2D, 3D training patches. Later these patches are employed in training the deep neural network. A sliding window approach as the name suggests a rectangular region i.e. patch size slides across each pixel on the testing slices and uses the trained model to carry out the pixel-wise classification.

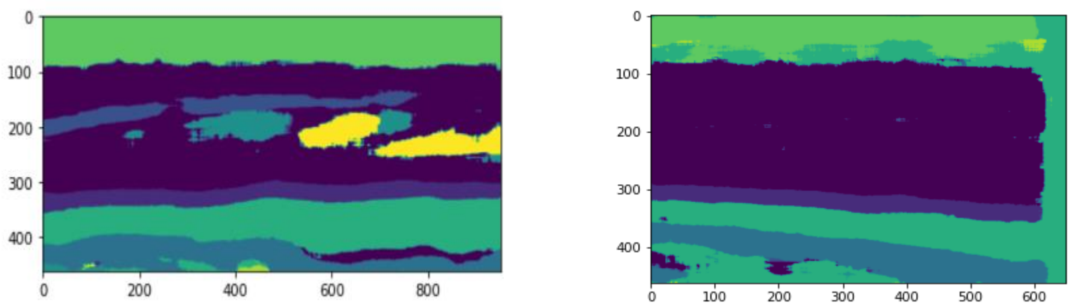

Figure 7- Patch extraction of crossline and inline images

4 Design Specification

A design workflow is incorporated with 3 stages, comprising of data preparation for the model, model implementation, and results desired as graphs, interpretation/visualization.

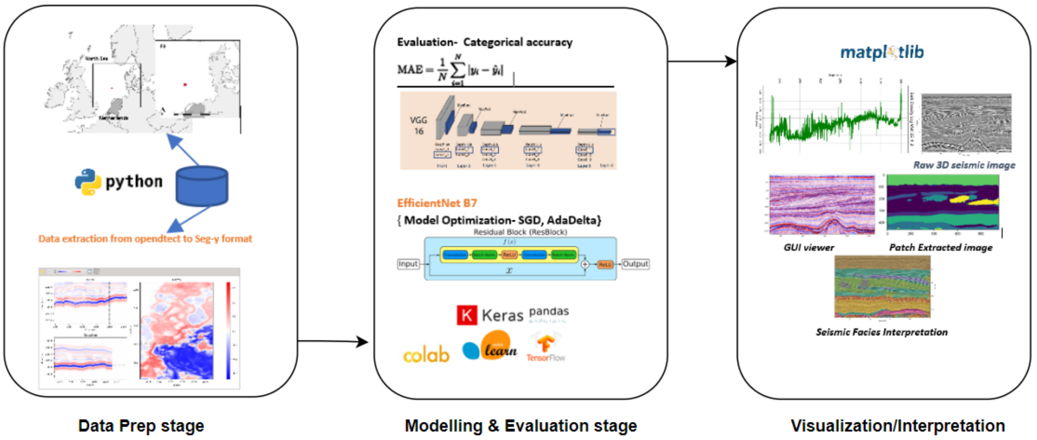

Figure 8-Seismic Image Classification- Design Flow

- The data preparation stage consists of all the methods performed for data gathering, extraction, feature selection, transforming, and enriching the dataset. The source of the data was exported from the OpenText repository and transformed into 3D seismic data using the seg.y GeoPackage database. Also, created an interactive visual tool to view the data and speed up the data analysis rather than manual as it is very important in data preprocessing the long recorded seismic data as each image in the dataset is of extreme size (600MB to 1GB).

- The modelling stage consists of an implementation of multiple convolutional neural networks such as VGGNet 16, ResNet, EfficientNet B7, and evaluated using MAE (Mean Absolute Error), categorical accuracy, precision. Fine-tuning parameters, model optimization are also carried out in this stage.

- The final stage is the result or the interpretation stage, where the output is desired to present in the graphs, plots for the visualization purpose.

Each model implemented has undergone optimization, cross-validation, and evaluated with the appropriate metrics which are discussed in detail in the following sections.

5 Implementation

5.1 Overall Workflow

The implementation of the research work is carried out in three phases namely, systematic data sampling, cross-validation, and fine-tuning parameters using optimization and regularization techniques. The main reason for choosing these techniques is lower processing time. In terms of implementation modes, foreseeing models that are trained with unique data of different significant formations, such as unconventional, faults, channels, salt deposition [21]. The key concerns correspond to the generalization error which generally sets the limits on how accurately a predicting model can predict unseen data. Finally, the exploration analysis workflows are concerned, one can assume that this technique will be used promptly after data acquisition, then trained models will be loaded from which interpreters will draw realizations thus performing validation and testing while incorporating their model modifications. In general, convolutional neural networks are typically built by a series of alternating convolutional layers and pooling layers; Initially, the image is fed into a convolutional layer, where convolutional kernels identify features and produce feature maps as output. After the convolution, the feature maps shrink spatially by the consecutive pooling layer. The CONV + POOL pattern is replicated numerous times in one series [5]. Finally, the performance of the last pooling layer is flattened and passed to many other fully connected layers to generate scores for different classes. The last fully connected layer commonly uses a softmax activation to seamlessly normalize the determined scores to the range [0, 1,2….8] and gives the prediction of probability for each class of the input image. All experiments are performed on 12GB NVIDIA Tesla K80 GPU and implementation is based on TensorFlow and Keras.

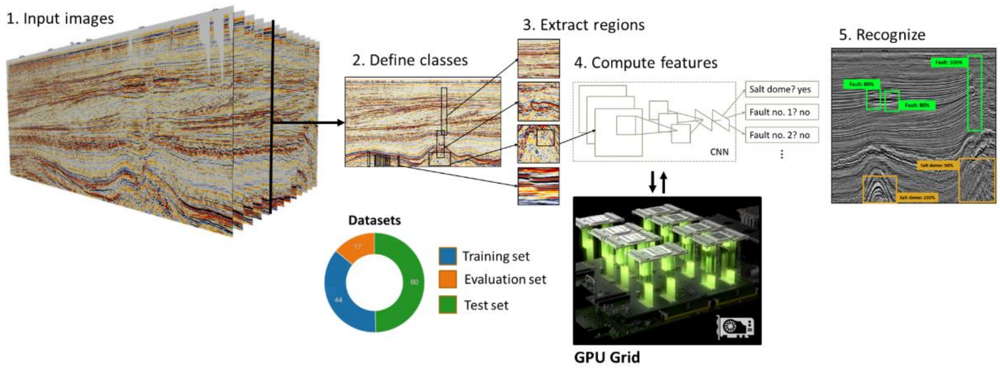

Figure 9 – Overall workflow

5.2 Training and Testing

The training phase has experimented with different settings for each algorithm experimented. The patch size was set to 64 to both VGGNet and EfficientNet models, 224 for the ResNet model. Accurate patch size is chosen to capture small range features and preserve the pixel location value. For training, only one type of slices has been used (inline slices with crossline combination), and validation is carried out on another type of slices or both inline and crossline which gives more adequate results. The similarity of adjacent slices of the same kind and the need to achieve a reproducible result motivated to follow a strategy for picking slices for training and validation, not by an arbitrary concept but evenly throughout the cube, i.e. so that the slices are as far apart as possible and thus cover the entire data variety. The training was conducted on the inline samples of the only 339th distributed on all the crosslines ranging from 330 to 1247.
For testing, inline 500th and 500th crosslines were used which is not considered in the training of any of the network performed. And 2 slices were used in the validation which is uniformly distributed adjacent to the training sample. In the following sections, the implemented model architectures, configuration settings are discussed in detail.

5.2 VGGNet16- Convolutional Neural Network

VGGNet-16 architecture is one of the excellent vision models proposed by K. Simonyan and A. Zisserman. The unique thing about VGGNet 16 is that instead of having numerous hyper-parameters, it has built with convolutional layers of 3x3 filter with stride 1 and using padding, max-pooling of 2x2 filter with stride 2 throughout the architecture, followed by 3 fully connected layers for classification which has made it popular in neural networks.
In this research work, the input to the network is of image dimension 64, batch size of 64, the sparse loss function is categorical cross-entropy with the mean absolute error loss function and the ReLu activation function which is computationally efficient and allows for backpropagation, with the last layers using softmax as the activation to output the probability for each class. Early stopping is used on the validation loss. The entire network is fine-tuned using the best parameters, replacement of SGD and AdaDelta optimizers with low weight decay, the low learning rate of 0.001 to update the weight parameters while minimizing the loss function, and Nesterov momentum acceleration optimization.

```
patch_size = 64 # for ResNet50 put 244, vgg -64
batch_size = 64
num_channels = 3
num_classes = 9
all_examples = 158812
num_examples = 7500
epochs = 3
steps=200
sampler = list(range(all_examples))
opt = 'adam'
lossfkt = ['categorical_crossentropy']
metrica = ['mae','acc',m]

from keras.optimizers import SGD
sgd = SGD(lr=1e-4, decay=1e-6, momentum=0.9, nesterov=True)
vgg.compile(loss=lossfkt,
                  optimizer=sgd,
                  metrics=metrica)
```

Figure 10- VGGNet using python- selected parameters

5.3 Residual Neural Network (ResNet 101/152)

The second network used in this study is the Residual Neural Network with 101 and 152 layers deep. The network has residual blocks with skip- or identity-connection which performs the identity mapping to add input x to the output after a few weight layers to solve gradient vanishing issues. Keras library in python is used to implement this model. The residual learning is easy to optimize and gain higher accuracy with increasing depths also lesser complex architecture with fewer layers than VGGNet. The configuration adopted in this research work is, the input image fed is of dimension 224 x 224, with learning rate 0.00, weight decay 0.0001, and Nesterov momentum 0.9. optimization was done using the Stochastic Gradient Descent with the nest parameters tuned in. In the experiments discussed shows that the testing accuracy deteriorated due to the skip connection where the useless information was transmitted from the shallow layers to the deep layers. ResNet shows better training results compared to other CNNs. Below is the chosen parameters for the baseline model of ResNets used for classification-

```
q = res_base.output
q = Flatten()(q)
q = BatchNormalization()(q)
q = Activation('relu')(q)
q = Dense(10,name = 'attribute_layer')(q)
q = BatchNormalization()(q)
q = Activation('relu')(q)
q = Dense(num_classes, name = 'pre-softmax_layer')(q)
q = BatchNormalization()(q)
q = Activation('softmax')(q)
resnet = Model(input=res_base.input, output=q)
resnet.summary()
```

Figure 11- Baseline classifier-model

5.4 EfficientNet B7

The EfficientNet B7 model is inspired by Tan and Le [14]. This model is particularly useful as it reduces the computational cost, battery usage, memory usage, and also training, inference speeds compared to other deep learning models. Using the compound scaling in the performance of the model tends to give the best model accuracy with computational efficiency. With a reduced number of parameter sizes and FLOPS, the model can achieve state-of-the-art accuracy higher than ResNet scaling models and VGGNet. Similar fine-tuning parameter settings as ResNet, VGGNet are even applied to EfficientNet B7. This model is applied to the seismic data with SGD optimizer, with weight decay 1e-6; batch norm momentum 0.9; learning rate 1e-3; dropout 0.5. Below is the summary of the EfficientNet B7.

```
top_bn (BatchNormalization)    (None, 2, 2, 2560)  10240    top_conv[0][0]
________________________________________________________________________________
top_activation (Activation)    (None, 2, 2, 2560)  0        top_bn[0][0]
________________________________________________________________________________
flatten_1 (Flatten)            (None, 10240)       0        top_activation[0][0]
________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 10240)      40960    flatten_1[0][0]
________________________________________________________________________________
activation_1 (Activation)      (None, 10240)       0        batch_normalization_1[0][0]
________________________________________________________________________________
attribute_layer (Dense)        (None, 10)          102410   activation_1[0][0]
________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 10)         40       attribute_layer[0][0]
________________________________________________________________________________
activation_2 (Activation)      (None, 10)          0        batch_normalization_2[0][0]
________________________________________________________________________________
pre-softmax_layer (Dense)      (None, 9)           99       activation_2[0][0]
________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 9)          36       pre-softmax_layer[0][0]
________________________________________________________________________________
activation_3 (Activation)      (None, 9)           0        batch_normalization_3[0][0]
================================================================================
Total params: 64,241,225
Trainable params: 63,909,987
Non-trainable params: 331,238
```

Figure 12- EfficientNet summary

6 Evaluation

To evaluate the models and assess the capability of the seismic facies prediction, classification (Precision), (Categorical accuracy), and regression (MAE) metrics are used. Precision to determine how close is the model's prediction to the observed values. And MAE that estimates the average error of the set of predictions. The experiments 6.1 and 6.2 are demonstrated by replicating the models implemented in [2] to improve the reliability, VGGNet, and ResNet are examined with the customized hyperparameter tuning and regularization techniques. In experiment 6.3 model scaling technique is redesigned following the work of [14] and implemented the EfficientNet b7 model.

6.1 Experiments with VGGNet-16

Experiment with VGGNet-16 is conducted by building 16 layers where the last three layers are fully connected layers for classification. With SGD and Adadelta optimizer and reLU (Rectified Linear Unit) activation function, the models were trained for epochs 10-15. The transfer learning method VGGNet-16 has experimented on the dataset: 1) Feature extractor- using a pre-trained model and 2) Fine-tuning process. The table-2 shows two different outcomes under two different optimizers are used. To enhance performance following fine-tuning is carried out in the experiment:

- The use of SGD, Adadelta are used to optimize and control the learning rate.
- And the regularization method- dropout is set to 0.5
- Cross-validation technique- Early stopping is used to immediately stop the training if the performance on the validation set gets worse.

While considering the accuracy, precision scores it is evident that the SGD performed better than the former Adadelta optimizer resulting in improved accuracy and precision overall. As the network fine-tuning process optimizes the weights in the layers responsible for classification, but also optimizes the weights that extract features from the input.

Table 2- VGGNet experiment result

Network-VGGNet-16	Accuracy	Precision	MAE
Fine-tuning (Stochastic Gradient Descent-optimizer)	97.1%	99.4%	0.085
(Adadelta-optimizer)	94.8%	99.1%	0.21

Based on the table-3 shown below, training accuracy increased efficiently from 93.3% to 96.2% using the optimized parameters, and also the loss reduced significantly by. With the default parameters, the reduction of loss and MAE took place slowly after several iterations (figure-13,14), whereas in the tuned parameters the reduction of loss and MAE is significant.

Table 3- train/test Accuracy

Experiment	Train Accuracy (%)	Test Accuracy (%)	Loss	Mean absolute error
Default Parameters	93.3%	94.8%	0.27	0.21
Tuned hyper parameters	96.2%	97.1%	0.094	0.085

In their work, Zini et al. [2] proposed a transfer learning approach that consisted of autoencoders to learn the source task and combined pre-trained VGGNet and they adopted an 8x8 sub-image input size and simple binary image output. For a fair comparison with their approach, the image size adopted in this work is 64x64, the batch size of 64, and fine-tuning parameters tested on VGGNet. Testing revealed an accuracy of 97.1 and MAE of 0.085 compared to 87.36% of accuracy reported in [2].

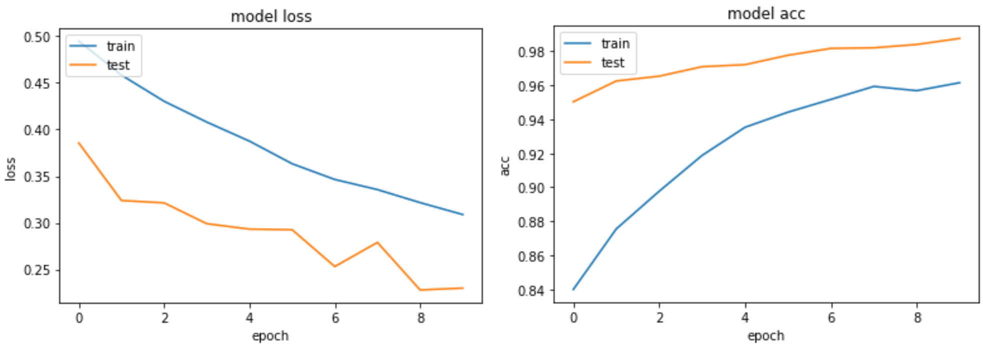

Figure 13- loss, and Accuracy of the model.

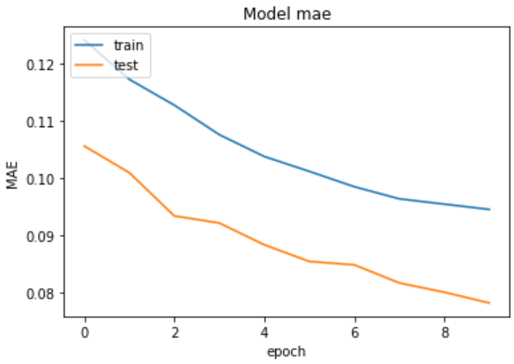

Figure 14- Mae of the model

Table 4- Results of fine-tuning methods performed on VGGNet

	Epochs	Train acc	Val acc	Mae	Precision	Loss
VGGNET- SGD	1	84.00%	95%	0.034	98.00%	0.73
	2	87.50%	96.30%	0.03	98.40%	0.67
	4	91.90%	97.1	0.025	98.70%	0.6
	6	94.40%	97.80%	0.023	99.00%	0.52
	8	95.90%	98.20%	0.022	99.30%	0.41
	10	96.20%	98.80%	0.02	99.40%	0.39
VGGNET- AdaDelta	1	34.70%	72.10%	0.175	88.80%	1.13
	2	62.70%	86.30%	0.149	95.10%	0.96
	4	82.10%	94.10%	0.126	97.70%	0.72
	6	88%	95.80%	0.114	98.50%	0.65
	8	92.30%	96.20%	0.108	98.90%	0.56
	10	93.30%	97.10%	0.102	99.10%	0.52

Initially, the experiment was conducted using the AdaDelta optimizer, with a default learning rate and a batch size of 32 for 10 epochs. As shown in table-4, the model resulted in 0.52 validation loss, 93.3% training accuracy and validation accuracy of 97.1% . While in the latter method (model with SGD, showed improvement with an increase in accuracy and reduction in validation loss and MAE, indicating the model can learn well.

The resulting prediction output of the VGGNet model is as below figure 4, indicating the facies classification in crossline and inline slices of the seismic cube.

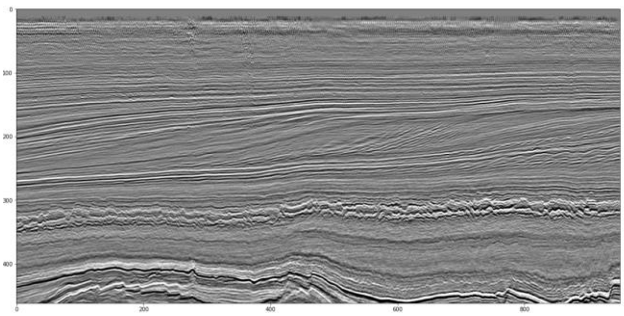

Figure 15- Raw F3 dutch seismic data

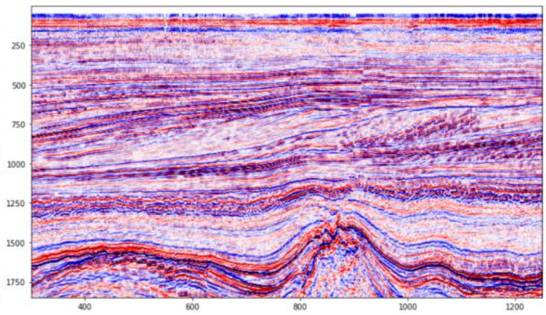

Figure 16- Processed seismic data using GUI

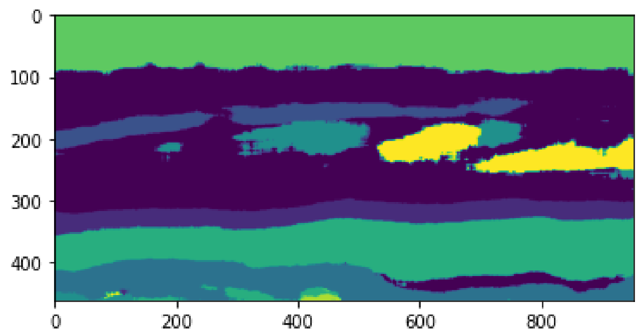

Figure 17- Patch extracted seismic data

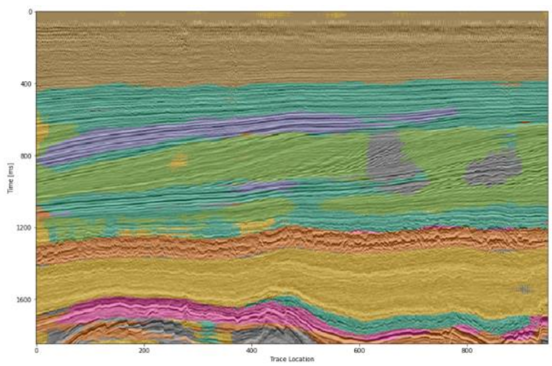

Figure 18- VGG 16 interpretation of seismic slice

6.2 Experiments with ResNet-101/152

The aim of this experiment is to target the number of parameters by trying to reduce it and to train the model much faster than the previous model by implementing multiple residual neural network models with varying the number of layers. To present a fair and direct comparison between different architectures of residual networks, taking into account of various performance indices useful for a comprehensive benchmark of ResNet models by measuring accuracy rate, memory usage, model and computational complexity.

The first ResNet model consists of 101 layers deep and the second ResNet model consists of 152 layers deep with 60 million parameters both. These residual neural networks utilize the skip connections to avoid vanishing gradient problem also to speed up the learning by skipping the layers.

The results of ResNet models are shown in the table below. The networking training indicates the perfect fit for the data, whereas the network performed poorly on the unseen data and hence the testing score is extremely least. The categorical accuracy recorded to ResNet 101 and ResNet 152 is 46% and 47% respectively (refer table-5). Also, the loss and MAE resulted in higher values than any other convolutional neural network experimented in this study. This shows that the ResNet is ineffective on transfer learning with pre-trained weights on the seismic data.

Table 5- ResNet result

Network	Training Acc	Test Acc	Categorical Acc	MAE	Precision	Categorical cross-entropy	Loss
ResNet 101	99%	0.096	46%	0.201	46%	16.78	23.83
ResNet 152	98.5%	0.085	47%	0.203	47.3%	10.29	25.49

However, the performance of ResNet 101/152 using the seismic data deteriorated in the testing score because of the skip connections of nonlinear processing. A potential explanation was that the efficient features in seismic data were sparse and distinct from computer vision images, so that useless information (noises) transferred from shallow to deep layers by the skip connections. ResNet filters poorly performed on the seismic data where it shows superior performance on the photography datasets.

In the work presented by Zini et al. [2], implemented ResNet with the Autoencoders as the source task, additionally generated the synthetic seismic data and overcame the model overfitting. Initially, the model showed lower accuracy of 24% and later method with the combination of autoencoders, fine-tuning resulted in 87.36% accuracy. This signifies that the functional capacities of autoencoder supported ResNet architecture with synthetic data performed far superior to the ordinary ResNet model.

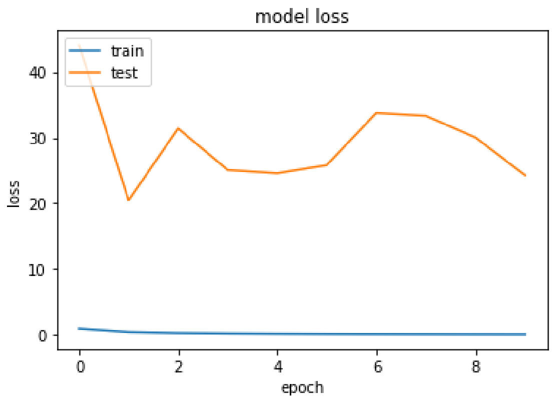

Figure 18- ResNet model loss

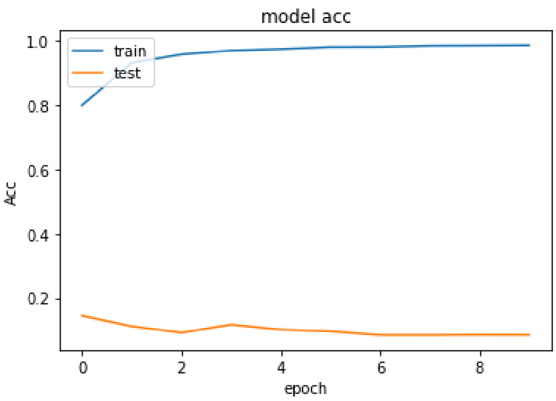

Figure 19- ResNet model accuracy

6.3 Experiment with EfficientNet- B7

To achieve the superior level performance with fewer parameters (66M) and lower the computational cost, the EfficientNet baseline model is applied on the dataset which is an 8.4x smaller model and 6.1x faster than ResNet. The EfficientNet B7 is imported from the Keras applications and loaded weights pre-trained on the ImageNet transfer learning data. In this experiment, the facies classification is considered from 0 to 9 but EfficientNet is built for 1000 class labels on the ImageNet data, which signifies the last few layers for classification is not used. Hence, the layers are excluded by specifying False to the include_top argument. And fine-tuning the network as follows-

- Adding the custom network layer on top of the baseline network.
- Freeze the baseline network.
- Training the part needed.
- Some layers are unfreezing in the base network.
- Combined training of these layers and the custom network.
- Customizing the behavior of the model parameters by specifying the decay 1e-6, batch norm momentum 0.9, learning rate 0.001, also linearly increasing dropout to 0.5.

The model was trained with a batch size of 64, SGD (Stochastic Gradient Descent) optimizer, and reLU (Rectified Linear unit) activation function for 10 epochs.

Table 6- EfficientNet experiment result

Network	**Loss**	**Categorical Accuracy**	**Accuracy**	**MAE**	**Precision**
Efficient B7	0.043	96.5%	99.7%	0.008	99.1%

Based on table-6 shown above, it indicates that the model achieved 99.7% accuracy and a decrease in loss, mae compared to the previous ConvNet (convolutional neural network). The simplicity of the EfficientNet B7 architecture with 66 million parameters made it quite appealing than the VGGNet-16 with 138 million parameters.

Tan et al. [14] presented their findings systematically demonstrating the EffcientNet B7 model on ImageNet and five other common transfer learning datasets and surpassing state-of-the-art accuracy of 93% to 98.9% on various transfer learning datasets with utilizing fewer parameters and outperforming ResNet models. To make a fair comparison, testing EfficientNet architecture on the seismic image yielded 99.7% transfer accuracy.

As shown in the figures below are the graphs of the model loss, accuracy, and mean absolute error (MAE). In the initial stage, the validation loss and MAE tend to fluctuate and after epoch 4 the values of loss and mae resulted stable with the right choice of the optimizer.

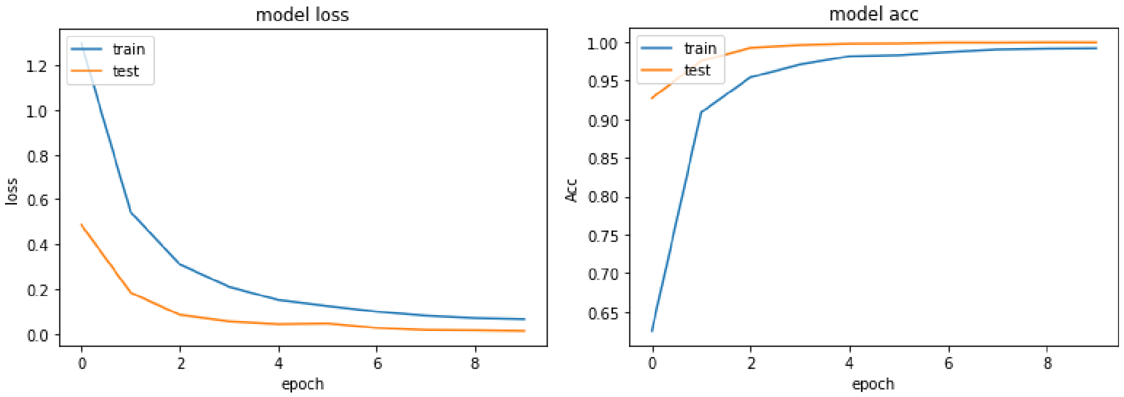

Figure 20- loss, and Accuracy of the model.

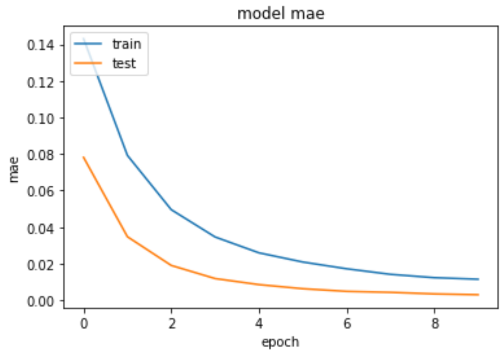

Figure 21- Mae of the model.

Table 7- Overall EfficientNet B7 performance

	Epochs	Train acc	Val acc	Loss	Mae	Categorical acc	Cat. cross entropy
EfficientNet B7	1	62.50%	92.70%	0.48	0.078	73.80%	1.72
	2	90%	94.80%	0.183	0.035	85.60%	0.8
	4	95.40%	95.44%	0.05	0.012	92.20%	0.44
	6	97.10%	97.11%	0.046	0.06	94.60%	0.31
	8	99.10%	99.60%	0.016	0.016	95.90%	0.24
	10	99.20%	99.80%	0.012	0.003	96.70%	0.19

7 Discussion

This study was empirically shown to perform well on classifying the seismic facies. Being trained on the large volume dataset of diverse seismic images, the research conducted has the potential to outperform other pre-trained networks. Three deep neural network models were implemented and evaluated in this study. Various techniques to enhance the model performance were adopted, such as data selection based on the influential attributes using GUI, feature extraction. In addition, substantial performance improvement is brought to the work, by optimizing the models using SGD, AdaDelta optimizers, and sophisticated regularization techniques such as dropout, early stopping, weight decay, batch normalization were employed. Up until now, the evaluation of individual ConvNets performance is assessed. And in this section, the discussion is made on how the conducted experiments outperform state-of-the-art work on two different applications: facies classification and model scaling. In [2] cast the bright spot detection as an image segmentation problem and used a basic version of VGGNet, ResNet with autoencoders as source task and the model showed the accuracy of 87.36% using VGGNet and ResNet initially performed poorly with 24% accuracy and later method with the combination of autoencoders, fine-tuning resulted in 87.36% accuracy. Our method outperformed the work presented by Zini et al. [2] by achieving 97.1% accuracy with lower MAE. In the case of ResNet did not yield notable results due to sparse seismic dataset. To overcome this issue, the model scaling method was employed by increasing the depth of the ResNet from 101 to 152 layers, which was unsuccessful due to skip-connection nature of ResNet filters poorly performed on the seismic data, where it shows superior performance in [2] by using synthetic simulated data. Another model scaling architecture called EfficientNet has experimented to achieve superior performance of 99.7% accuracy, categorical accuracy of 96.5%, and MAE of 0.008 by employing only with fewer parameters in comparison to ResNet models. While comparing the results obtained in this experiment with the [14] resulted in 98.9 as the top accuracy performed on popular transfer learning datasets.
The performance of Residual Neural Network was very poor resulting in lower accuracy and an extremely higher error rate which is due to skip connections characteristic, sparse seismic data. These complexities have impacted several seismic facies analysis studies in the past where the models resulted

in falsified classification. Finally, VGGNet and EfficientNet can be considered as the most suitable models in seismic data analysis problems.

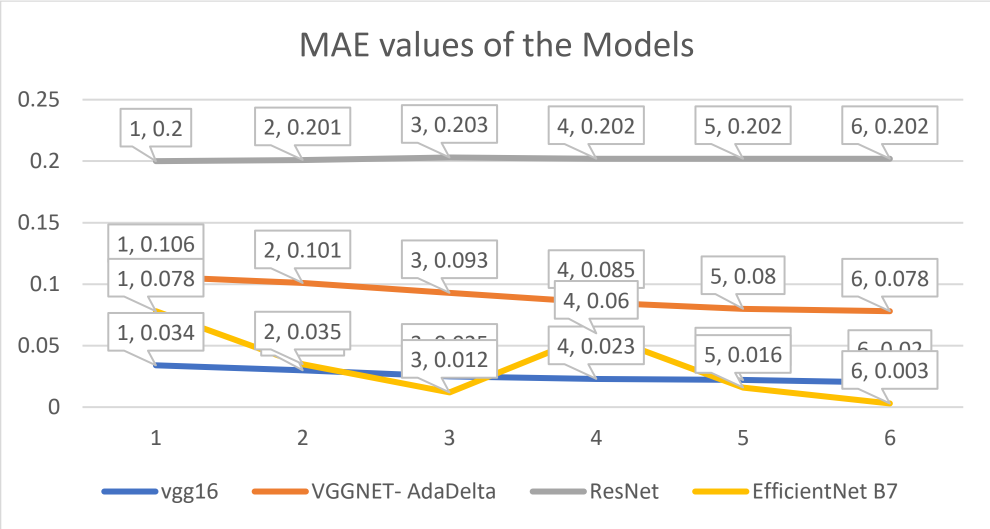

Figure 22- MAE values for VGGNet, ResNet, EfficientNet

8 Conclusion and Future Work

The presented novel deep neural network approach on seismic data is promising and feasible for the seismic facies classification. Transfer learning applied on ConvNets demonstrated that knowledge gained by training a network with a particular collection of seismic data could be reused in a similar task which means once the model is trained, one can use the parameter values as the starting point to apply the same model to a different dataset. ConvNets contemplates the spatial aspect of the seismic images and exhibits hierarchical structures for identifying complex features through layers of convolutional nodes. During the training phase, the filter weights are optimized and regularization techniques were introduced to make the model simpler and avoid overfitting. The proposed method is pragmatic, particularly for researchers and interpreters struggling to obtain a large amount of labelled real data. Additionally, there are two obvious benefits: one is its cost-effectiveness, that is to say, many labelled parts can form a network that can make precise predictions; another benefit is that it enables to use real data as a training set without being concerned about insufficient data. The pre-trained networks VGGNet, EfficientNet showed successful outcomes in different aspects: VGGNet being most accurate in classifying the seismic facies with a lower rate, EfficienNet benefits in lower computational cost with only a few parameters. Furthermore, the proposed model can predict 3D seismic sections of any volume, and after determining several parameters, the entire prediction is automated. Interpreters only need to select and interpret some parts when analyzing a new seismic volume, after which the model will predict faults in any other region of the volume. Instead of the high labor and time costs needed to manually mark faults, the proposed method will significantly increase the efficiency of interpretation. Future studies will assess the performance of the proposed method on more complex seismic data and analysis issues such as waveform-based earthquake prediction, the efficacy of transfer learning on other forms of deep neural networks on seismic data analysis, and performance on a larger volume of datasets with varying noise rates.

Acknowledgment

Foremost, I would like to express my sincere gratitude to my supervisors Dr. Paul Stynes, Dr. Pramod Pathak for the continuous support, insightful comments, and hard questions throughout the research work. For 13 weeks, my supervisors extended their valuable assistance to the success of this study.

Thanks to dGB Earth Sciences for enabling the F3 North Sea seismic data to the public and my appreciation also extends to my family for their exceptional love, constant understanding. I also acknowledge thanks to my friends who always inspired me with positive motivations.

References

[1] "Detecting geological structures in seismic volumes using deep convolutional neural networks", Publica.fraunhofer.de, 2020. [Online]. Available: http://publica.fraunhofer.de/dokumente/N-507059.html. [Accessed: 13- Aug- 2020].

[2] J. el Zini, Y. Rizk, and M. Awad, "A Deep Transfer Learning Framework for Seismic Data Analysis: A Case Study on Bright Spot Detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3202–3212, May 2020, doi: 10.1109/TGRS.2019.2950888.

[3] Y. Zeng, K. Jiang, and J. Chen, "Automatic Seismic Salt Interpretation with Deep Convolutional Neural Networks," Nov. 2018, doi: 10.1145/3325917.3325926.

[4] Y. Huang *et al.*, "GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism," Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.06965.

[5] M. Araya-Polo, T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, "Automated fault detection without seismic processing," Leading Edge, vol. 36, no. 3, pp. 208–214, Mar. 2017, doi: 10.1190/tle36030208.1.

[6] S. Jetley, N. A. Lord, N. Lee, and P. H. S. Torr, "Learn To Pay Attention," Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.02391.

[7] H. Di, "Tu D 11 Seismic Fault Detection from Post-Stack Amplitude by Convolutional Neural Networks."

[8] 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, Aug. 2019, doi: 10.1190/int-2018-0235.1.

[9] A. U. Waldeland, A. C. Jensen, L.-J. Gelius, and A. H. S. Solberg, "Convolutional neural networks for automated seismic interpretation," *The Leading Edge*, vol. 37, no. 7, pp. 529–537, Jul. 2018, doi: 10.1190/tle37070529.1.

[10] D. Chevitarese, D. Szwarcman, R. Mozart, D. Silva, and E. V. Brazil, "Transfer Learning Applied to Seismic Images Classification*," 2018, doi: 10.1306/42285Chevitarese2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." [Online]. Available: http://image-net.org/challenges/LSVRC/2015/.

[12] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," May 2016, [Online]. Available: http://arxiv.org/abs/1605.07146.

[13] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.04861.

[14] M. Tan and Q. v. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946.

[15] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *Lecture Notes in Computer Science*, pp. 234-241, 2015. Available: 10.1007/9783319-24574-4_28 [Accessed 13 August 2020].

[16] A. Milosavljevic, "Identification of salt deposits on seismic images using deep learning method for semantic segmentation," *ISPRS International Journal of Geo-Information,* vol. 9, no. 1, Jan. 2020, doi: 10.3390/ijgi9010024.

[17] 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, May 2019, doi: 10.1190/geo2018-0646.1.

[18] C. Gu, Y. M. Marzouk, and M. Nafi Toksöz, "Bayesian deep learning and uncertainty quantification applied to induced seismicity locations at the Groningen gas field in the Netherlands-What do we need for safe AI?"

[19] T. Zhao, "Seismic facies classification using different deep convolutional neural networks," in *2018* SEG International Exposition and Annual Meeting, SEG 2018, 2019, pp. 2046–2050, doi: 10.1190/segam2018-2997085.1.

[20] F. Qian, M. Yin, X.-Y. Liu, Y.-J. Wang, C. Lu, and G.-M. Hu, "Unsupervised seismic facies analysis via deep convolutional autoencoders," 2018. [Online]. Available: http://library.seg.org/.

[21] M. Araya-Polo, J. Jennings, A. Adler and T. Dahlke, "Deep-learning tomography", The Leading Edge, vol. 37, no. 1, pp. 58-66, 2018. Available: 10.1190/tle37010058.1 [Accessed 16 August 2020].