

Introduction:

This report is a project plan for Zongpeng Chen's Independent Research Project (IRP) on discovering the feasibility of applying CNNs on supervised seismic facies and other structures identification and classification. It contains 7 parts including the rationale and project objectives, the data set used in this project, a literature review, proposed approach, list of milestones, conclusion and a list of key references.

Rationale and Project Objectives:

Convolutional Neural Networks (CNNs) as one of the most popular deep learning methods in these years, is widely used for image classification (Goodfellow et al., 2016). Different from traditional feedforward neural networks which are applied to some distinctive features, CNNs are applied directly to the input image (van der Baan, 2000), thus accounting for the spatial organization of the image. They usually reduce the original dimension of the input image and transfer the information of the input image to some subsequent layers of the network (Szegedy, 2015). With more spatial information stored and used, the quality of models trained with CNNs is proved to be more advanced than those trained with traditional feed forward neural networks.

A seismic section is essentially an image of filtered reflectivity, which displays the different seismic wave reflections into an image from which features can be retrieved by human interpreter or by a machine. The different reflection features usually relate to different seismic facies and structures, which are interpreted by expert geoscientists. It usually takes several months for an expert's team to accurately interpret all seismic images of a field.

As more successful applications of CNNs on image feature identification happen in different areas, more geoscientists now consider using CNNs for solving geoscientific problems (Waldeland, 2018). In concept, CNNs can help with nearly every kind of image feature classification problem, so it is rational to test this method on seismic image interpretation which is essentially an image classification problem as.

This project is designed to discover the feasibility of applying CNNs on supervised seismic facies and other structures identification and classification to reach the interpretation result of human as shown in Figure 1 but faster and with a very limited set of training image.

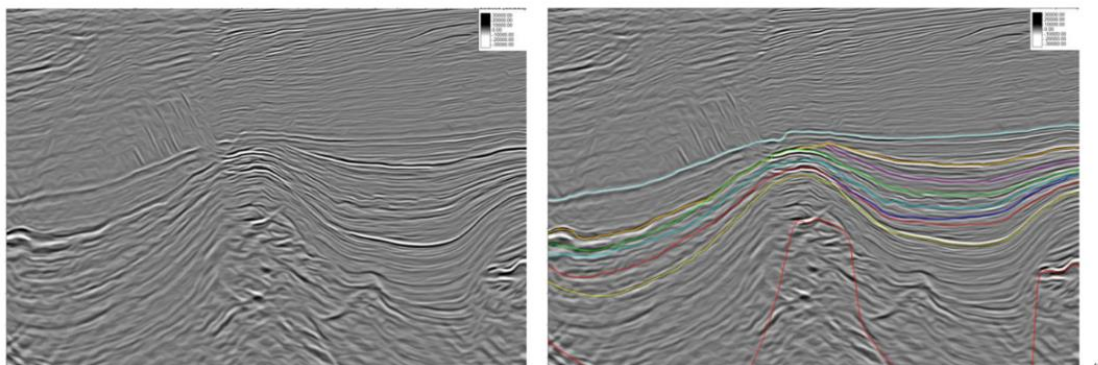


Figure 1. (a) Example seismic section. (b) Same section after interpretation by human.

The main task of the project is to segment 2D and 3D image patterns using Convolutional Neural Networks on massive set of SEG-Y format data (seismic image/data) over one field. I will also test the effectiveness of traditional Feed-Forward Neural Networks on doing the same task.

In supplement, I will generate realistic synthetic seismic images using Cycle Generative Adversarial Networks (CycleGANs) (Goodfellow, 2014 and Zhu, 2017) with the seismic images and some real geological outcrops for research purposes.

Data Set:

The training field has chosen to be Gonelle section in Gabon, Africa. All the SEG-Y data for this project is owned and provided by Perenco. 5 IL (In-lines) sections are used to train, validate and test the algorithms.

In total, Gonelle is covered by 1100 IL (In-lines) and 1400 CL (Cross-lines).

Literature Review:

There are already some works done by other groups on attempting CNNs on seismic images. One of the well-known groups is led by Dr. Waldeland, who published ‘Convolutional neural networks for automated seismic interpretation’. In this paper, he shared a basic methodology about this task (Waldeland, 2018). Classifying seismic facies in seismic images is a little bit different from the general CNNs image identification process. For example, when dealing with MNIST database, each sample is regarded as a single object. A sample from MNIST can be classified as a number between 0 to 9, but not the combination of them. However, with seismic images, each different sample of image will contain different classes of object. A part of the image could be a layer of turbiditic sandstones package while the other part of the image may be a shale package. We need a way to better slice the original images and train the model with some small objects to distinguish all the different seismic events. In his article, Dr. Waldeland trains a neural network on a 3D dataset acquired from the Barents Sea. With his focus on salt identification, he classifies each single pixel in the whole datasets as just ‘salt’ or ‘not salt’. Then he regards each pixel separately and selects a $65 * 65 * 65$ cube around each pixel as single training datapoint. The corresponding label is the label of each center pixel. We can train with different cube size to fit our own needs. The next important thing of this task is the construction of the Neural network. Dr. Waldeland in this article use a network configuration presented in Figure 2 (Waldeland, 2018).

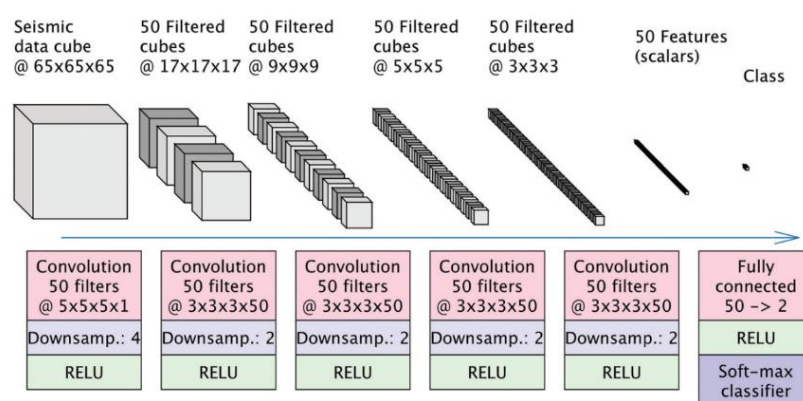


Figure 2. The neural network used for salt picking.

After testing the number of layers, number of nodes in each layer, Pooling, dropout and different activation functions for the network architecture, he comes out with this Network configuration which has 5 convolutional layers and 1 fully connected layer. As claimed in the article, none of the above conditions he has changed affects the results much, so he just implements batch norm and data augmentation to improve the model accuracy. The network is trained on one manually labeled inline slice (Figure 3a) and the result testing on another inline section is shown in Figure 3b (Waldeland, 2018).

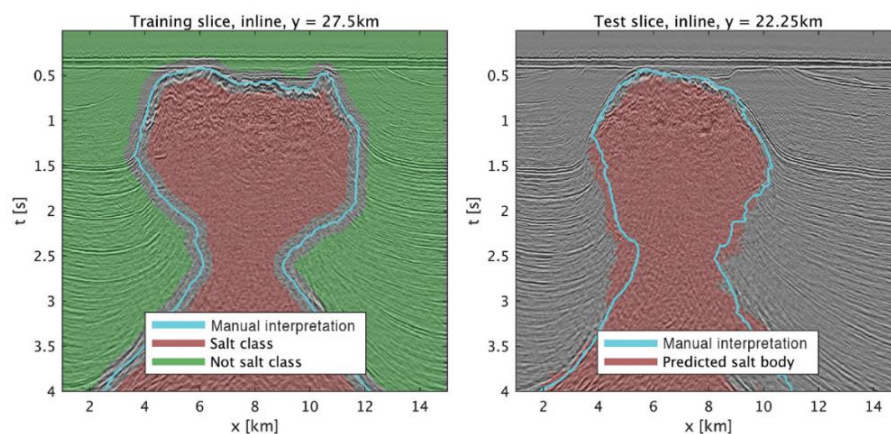


Figure 3. (a) The inline slice used for training. (b) The test slice.

With basically the same data type and similar target, it is interesting to try to classify seismic facies and other geological structures using CNNs on the Gonelle data set. And as suggested in this article, training with more inline slices will improve the model (Waldeland, 2018). For my project particularly, I will also compare the effectiveness of this neural network trained both on 2D and 3D while trying to keep the training images to the minimum.

Approaches:

All the data provided by Perenco will be in SEGY format, which is the standard format used for storing geophysical data. An external library need be installed to load the SEGY data in Python. Before loading the data, we need to define how many classes does the 2D reflectivity-based images contain and provide each pixel with the corresponding class label (7 labels here, one for each geological package) (Figure 4). Both slices are SEGY files originating from a professional geophysical software namely OpendTech®

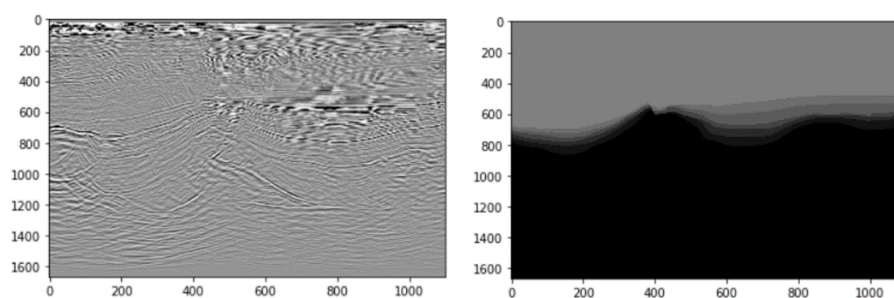


Figure 4. (a) Original seismic data slice. (b) Class label slice.

After loading all the SEGY files in Python, checking that there is no data loss, we need to split the original SEGY files into training, validation and test datasets. I will use the standard 3:1:1 to split the original datasets which means that the training set constitutes sixty percent of the total original datasets while both the validation and test sets constitutes twenty percent respectively. For example, if we have 5 whole seismic slices, we should select carefully 3 whole slices as training sets and 1 different slice for each validation and test set, although the training happens on mini batches around pixels. The seismic slices should be selected as separate as possible covering the field to avoid high correlation between each slice. I will use k-fold method to validate the model as well.

After that, we need to pad the seismic slice. In Dr. Waldeland's paper, he selects a $65 * 65 * 65$ cube around each pixel as a single training object. However, a pixel near the edge does not have enough neighbors. I will pad 0's for both 2D and 3D seismic slices because edge itself is a feature we need to keep.

Next, we need to segment the seismic slice according to the mini batch size. To avoid memory problem, I will only record the coordinate of each pixel. Another step needed to do is balancing the data. We need to make sure that each class has basically the same number of training objects. In case of that, I will set a standard for each of the classes. When a class has more training objects than the standard, it will randomly abandon some of the training objects until the number of them gets close to the standard. Relatively, a class that does not have enough training objects will randomly append new training objects using data augmentation: new created objects will be the duplication of existing objects but with some slight modifications. After normalization, the balanced dataset will be ready to use.

To train the model through Pytorch, we need to first transform the data to Pytorch Tensors. A custom Pytorch Dataset is needed to return actual images from the coordinates. For data augmentation, the training objects in each epoch will have some chance to transform. I will only use rotation between 10 degrees for this task because of the geology. This function will be operated through the custom dataset as well. The labels will always be unchanged. Similarly, the validation and test sets need to be fed in the custom Pytorch Dataset as well, but no transformation needed at this time.

After loading the datasets to the Pytorch Dataloader, the neural networks need to be set. As stated in the project objectives, I will use both Feed-Forward and Convolutional Neural Network, compare their results and then use Cycle Generative Adversarial Networks (CycleGANs) to produce synthetic seismic images with respect to real geological outcrops (Zhu, 2017). So, six different neural networks are needed in total (two discriminator networks and two generator networks for CycleGANs). After construction of the prototype networks, different layer types, layer numbers and node numbers need to be tested by comparing the results of training and validation sets to get us the best neural network architecture. I will use both accuracy and weighted accuracy as metrics for this model coming with

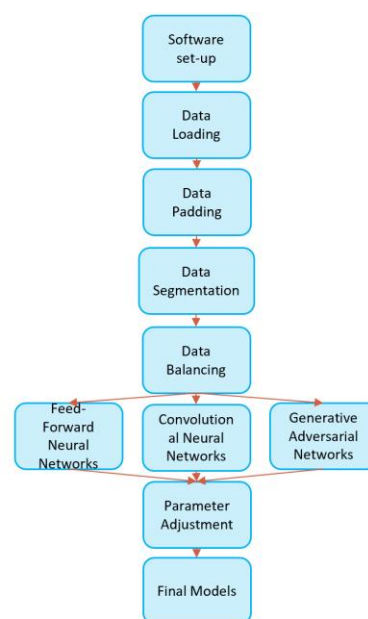


Figure 5. Workflow of the approach

a Confusion Matrix. Same sensitivity and uncertainty tests are also performed on hyper-parameters, optimizers and loss functions. The whole workflow is showing in Figure 5.

I will try other approaches to improve the model. At first, I will investigate Unet (Ronnerberger, 2015) which doesn't need data segmentation (Zhao, 2018). And on the other aspect, I will use other seismic attributes besides reflectivity to train as well (Wrona, 2018). I will first add the attributes to the 2D training set to make it 3D and ultimately, I will try to include them in the 3D training set for a 4D training.

In the end, the models should be implemented on the test set and at last on the full field of Gonelle to give us the final performance and a sense of their ability to generalize.

Milestones:

Milestone no.	Sub Task no.	Milestone name	Date due
0		IRP kick off	03.06.2019
1		Project Plan	28.06.2019
	1.1	Requirements Analysis	11.06.2019
	1.2	Review of Current Theory	05.07.2019
	1.3	General data Preparing, Labelling, Processing and Loading	21.06.2019
	1.4	High-Level Code Structure Design	21.06.2019
	1.5	Project Plan	28.06.2019
2		Feed-Forward model	21.06.2019
	2.1	Data preparation	11.06.2019
	2.2	First Prototype Program	17.06.2019
	2.3	Final model	20.06.2019
	2.4	Results compiling	21.06.2019
3		CNN model	02.08.2019
	3.1	Data preparation	04.07.2019
	3.2	First Prototype Program	12.07.2019
	3.3	Final model	26.07.2019
	3.4	Results compiling	02.08.2019
4		CycleGAN model	16.08.2019
	3.1	Data preparation	31.07.2019
	3.2	First Prototype Program	06.08.2019
	3.3	Final model	09.08.2019
	3.4	Results compiling	16.08.2019
5		Final report and Presentation	30.08.2019
	5.1	Final results compiling	23.08.2019
	5.2	Outline of Report	23.08.2019
	5.3	Github Repository	23.08.2019
	5.4	Final Draft of Report	30.08.2019
	5.4	Presentation	mid-September

Conclusion:

This project plan summarizes the basic process I will use in my Independent Research Project to discover the feasibility of applying CNNs on supervised seismic facies and other structures identification and classification. The results and further discussions will be introduced in my final report.

Key References:

- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014, Generative Adversarial Networks, arXiv:1406.2661.
- Goodfellow, I., Y. Bengio, and A. Courville., 2016, Deep Learning: MIT Press.
- Ronneberger, O., Fischer, P., and U. Brox, 2015, U-Net: Convolutional networks for biomedical image segmentation: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, 234-241, https://doi.org/10.1007/978-3-319-24574-4_28.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., 2015, Going deeper with convolutions: IEEE Conference on Computer Vision and Pattern Recognition, <https://doi.org/10.1109/CVPR.2015.7298594>.
- Van der Baan, M., and Jutten, C., 2000, Neural networks in geophysical applications: Geophysics, 65, no.4, 1032-1047, <https://doi.org/10.1190/1.1444797>.
- Walderland, A. U., and Solberg, A. H. S., 2017, Salt classification using deep learning: 79th Conference and Exhibition, EAGE, Extended Abstracts, <https://doi.org/10.3997/2214-4609.201700918>.
- Walderland, A. U., Jensen, A. C., Gelius, L., Solberg, A. H. S., 2018, Convolutional neural networks for automated seismic interpretation: The Leading Edge 2018 37:7, 529-537, <https://doi.org/10.1190/tle37070529.1>.
- Wrona, T., Pan, I., Gawthorpe, R. L., and Fossen, H., 2018, Seismic facies analysis using machine learning: Geophysics 2018 83:5, O83-O95, <https://doi.org/10.1190/geo2017-0595.1>.
- Zhao, T et al., 2018, Seismic facies classification using different deep convolutional neural networks, Seismic facies analysis using machine learning: SEG Technical Program Expanded Abstracts 2018. August 2018, 2046-2050, <https://doi.org/10.1190/segam2018-2997085.1>.
- Zhu, J., Park, T., Isola, P., Efros, A. A., 2017, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, arXiv:1703.10593.