Multi-Class Segmentation of High Resolution Geophysical Data Using Deep Learning

Cicero Pereira Batista Junior*[1] and Orlando Demetrio Zaloti Junior[1], [1]Petrobras

Abstract

Geohazard studies aid the identification of geological features on surface and subsurface that might be related to potential risks for the installation of subsea engineering projects in the oil and gas (O&G) industry. The detection and interpretation of these risk features are crucial tasks both in the exploration and development phases of oil production. Within this context, high resolution geophysical methods are used to manually identify and map the features of potential hazards to subsea installations. This work presents the application of computational techniques to raise the level of automation in geohazard studies, using models based on deep learning through U-Net, a type of Fully Convolutional Networks (FCNs). The result was the implementation of an intelligent system for automating the task of identifying and classifying geohazards, which makes it possible to mitigate the time spent in the search for occurrences of potential risks and provides greater attention to their analysis.

Introduction

Subsea engineering projects in the O&G industry require several input products, including geohazard studies.

Geohazard is defined as "A geological state, which represents or has the potential to develop further into a situation leading to damage or uncontrolled risk" (<https://www.ngi.no/eng/Projects/ICG-International-Centre-for-Geohazards/Offshore-Geohazards>. Access: June/2021). I.e., they are all the geological elements of the seabed potentially causing any kind of damage to people, the environment or underwater infrastructure. The following characteristics can be cited as geohazard features: slope instability processes, high slope gradients, geological faults and the presence of carbonates (possible corals). Hence, geohazard studies are paramount in the initial phases of subsea engineering projects as they will be able to identify potential risks to the installation of pipelines, equipment, wells, mooring systems, among others. Geohazard investigations in the marine environment can be carried out indirectly through the acquisition, processing and interpretation of high resolution geophysical data, including SONAR (Sound Navigation and Ranging).

SONAR is a method based on the principles of sound propagation underwater, by the emission of a high frequency acoustic signal at regular time intervals by two submerged transducers (emitters and receivers), which points to both sides of the surface, in relation to the course of navigation of the vessel. This method allows obtaining detailed information through different signal reflection patterns, enabling the geohazard features mapping, the type of seabed (sands, muds, rocks, carbonates/corals, etc.), inference as well as anthropogenic structures (pipelines, equipment, scraps, wrecks, etc.) identification, as illustrated in Figure 1.
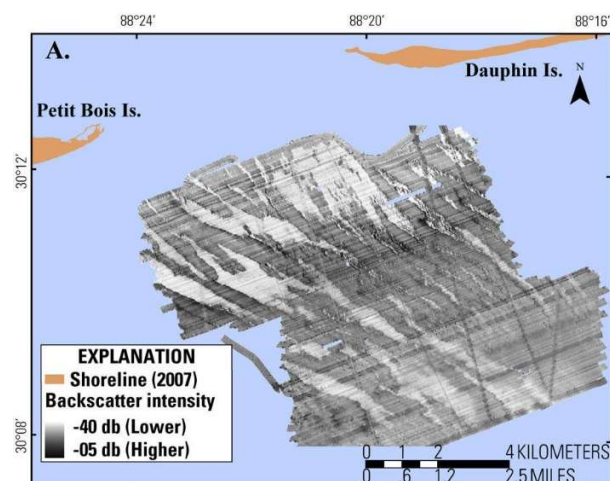

Figure 1 – *SONAR image example. Extracted from: <https://www.usgs.gov/media/images/side-scan-SONAR-mosaic-offshore-petit-bois-island>. Access: June/2021.*

Currently, the detection and interpretation of geohazard features through SONAR data is performed manually, requiring a considerable amount of time by specialists. To automate the identification of such features, this work presents the application of deep learning techniques using Convolutional Neural Networks (CNNs) models, specifically U-Net type of Fully Convolutional Networks (FCNs).

According to Vargas et al (2016), CNNs are currently widely used mainly in the computer vision area, such as in classification, detection and recognition image and video data. Typically, CNNs are composed of a set of convolutional and pooling layers that extract representative features while reducing the dimensions of the input images. Afterwards, the final characteristics are passed through some fully connected layers and an activation layer to get the final prediction (Figure 2). However, the use

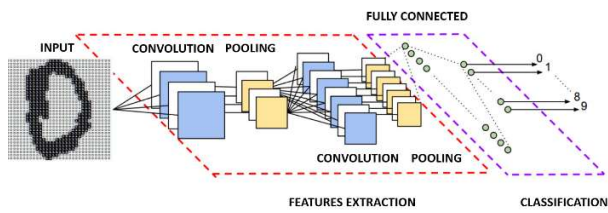of fully connected layers involves a loss of spatial information as well as a restriction on image input size.

Figure 2 – *The overall architecture of the Convolutional Neural Network, Vargas et al (2016).*

The approach of an FCN was introduced by Long et al (2015). In this work, the authors propose an adaptation of existing CNNs, transforming the fully connected layers into convolutional ones, allowing the generation of segmentation feature maps for each image. Regarding the segmentation task, it is necessary that the neural network combines the location information with the contextual information of the image feature maps to be predicted. Among the FCNs, U-Net has been proven to be effective especially for image segmentation. According to Ronneberger et al (2015), the U-Net network consists of two paths: contraction (encoder) and expansion (decoder). The contraction path can gather contextual information through sequences of convolutional layers and maxpooling. The expansion path can obtain the locational information by increasing the output of the contraction path to its original size by sets of transposed convolutions followed by convolutional layers. After each transposed convolution, the result is concatenated with its corresponding contraction path (Figure 3).
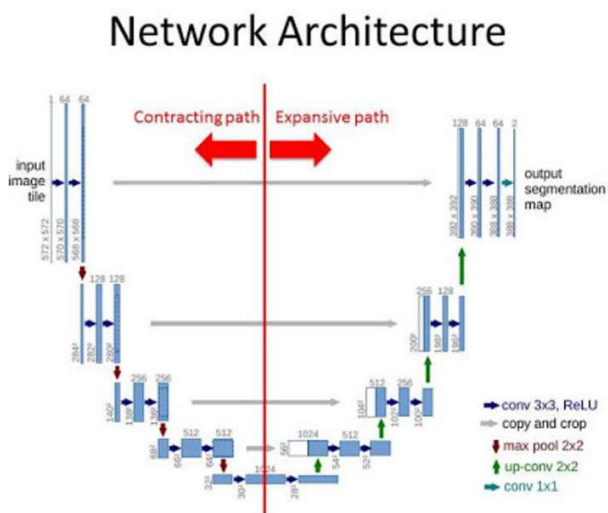

Figure 3 – *Architecture U-Net, Ronneberger et al (2015).*

Method

The workflow adopted to perform the semantic segmentation of SONAR images into two distinct classes of geohazard features, namely mud and carbonate (possible coral) is presented below, using supervised learning through U-net neural networks, Figure 4.
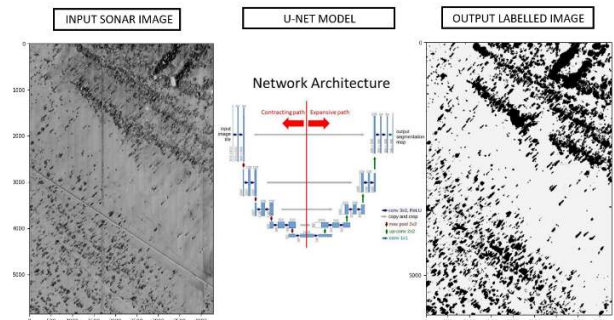

Figure 4 – *Global methodology to detect geohazards from a set of SONAR image using a U-Net architecture.*

In the SONAR image of Figure 4, the carbonate are the features with the higher amplitude (in black), while the mud shows lower amplitude (in grey).

There are mainly four phases in the workflow, (i) data acquisition, (ii) pre-processing, (iii) training and (iv) testing (Figure 5).
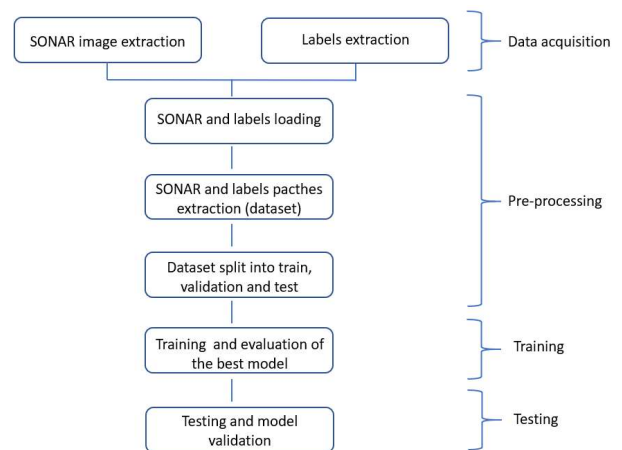

Figure 5 – *General workflow chart used in this work.*

(i) Data acquisition

The SONAR image, manual segmentation and classification of geohazard features were extracted from Petrobras' on premise database. In all the following stages, Python language (version 3.8) was used and the TiffFile, Scikit-learn, Matplotlib, NumPy, Skimage, OpenCV, TensorFlow and Keras libraries were used. The settings of

the hardware resources used were Intel(R) Core(TM) i7 CPU @ 1.99Ghz RAM 8GB.

(ii) Pre-processing

At this stage, the SONAR images and their labels were loaded with the following shape: 5854m x 4288m x 3 and 1m x 1m resolution. The patches for both were extracted setting the size to 128 x 128 x 3 (16416 patches). Afterwards, the pixel values of the images were converted to floating-point type. Then, the data was normalized by the highest pixel value. Finally, data were splited into training (~70%), validation (~15%) and testing (~15%). The result is shown in Figure 6.
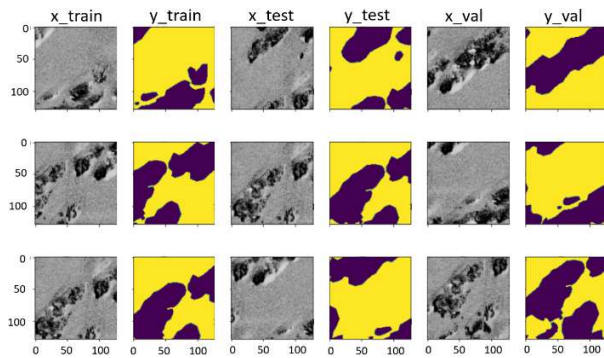

Figure 6 – *Train, validation and test data.*

(iii) Training

The U-Net architecture is composed of four convolutional blocks in the contraction path, where each block has two 2D convolutional layers and a 2D max pooling layer. Before beginning the expansion path, two more 2D convolutional layers are added. Then, the output of the contraction path is upsized by four more blocks with transposed convolutional layers. At the end, a sigmoid activation function is used to obtain the final segmentation.

The training (x_train, y_train) and validation (x_val, y_val) data were used in this phase, in addition to Early Stopping regularization method to avoid overfitting.

The best model with the best performance was saved to be loaded in the testing phase.

(iv) Testing

Here, the best model of the training phase was loaded and the accuracy and loss in the test data (x_test, y_test) were evaluated.

Results and Discussion

Several hyperparameters were manually tested and adjusted in the training phase, aiming to improve the model performance. Table 1 presents the best fits of the

hyperparameters, while Table 2 presents the performance of the best U-Net model.

Table 1 – *Best hyperparameters in U-Net model.*

Build U-Net Model				
input shape = (128,128,3)				
Contraction path				
Conv2D layers				
filters	kernel size	padding	kernel initializer	activation
16, 32, 64,128, 256	3,3	same	he normal	relu
MaxPooling2D layers				
pool size			strides	
2,2			2	
Expansive path				
Conv2DTranspose layers				
filters	kernel size	padding	kernel initializer	activation
128, 64, 32,16	2,2	same	he_normal	relu
Conv2D layers				
filters	kernel size	padding	kernel initializer	activation
128, 64, 32,16	3,3	same	he normal	relu
output shape = (128,128,1)				
Conv2D layer				
filters		kernel size		activation
1		1,1		sigmoid
Compile Model				
optimizer		loss		metrics
adam		binary cross entropy		accuracy
Fit model				
callbacks			batch size	epochs
Model Checkpoint (save best only)	Early Stopping		16	10

Table 2 – *Model performance during the training phase.*

Best U-Net model			
Train data		Validation data	
accuracy	loss	accuracy	loss
0.9980	0.0048	0.9979	0.0052

Figures 7 and 8 show the evolution of accuracy and loss, respectively, during the training phase.
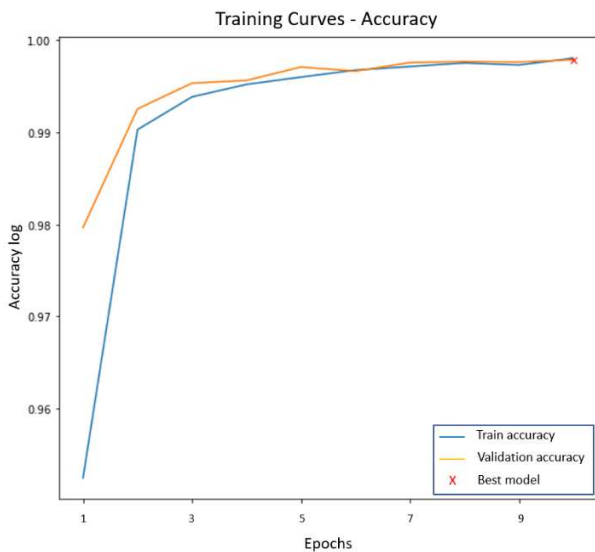

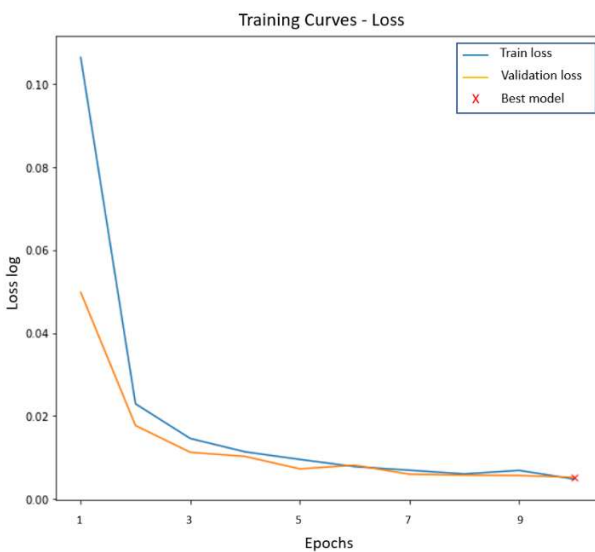*Figure 7 - Training curves – accuracy.*

Figure 8 – Training curves – loss.

Despite few training periods (10 epochs), it was observed that the model reached 99% accuracy in the second epoch and from then onwards, it improved more slowly until epoch 10, when it reached the best accuracy of 99.8%. It is noteworthy that for the validation data, the accuracies were higher than for the training data until epoch 5, when they converged until the end of the training.

In relation to the loss value, there was an abrupt drop also in the second epoch and from then on, the model improved more slowly until epoch 10. Confirming the trend of what happened with the accuracy values, the loss values were higher in the training data up to epoch 5, when they converged until the end of training.

After the training phase, test data (x_test, y_test) were provided to the best U-Net model from the training phase to verify its generalization capability.

Table 3 shows the performance of the U-Net model and it is noted that the model was able to learn when new data were presented.

Table 3 – Model performance in the test phase.

U-Net model	
Test data	
accuracy	loss
0.9979	0.0051

Figure 9 shows the comparison between the output test data (y_test) and the data predicted by the U-Net model.
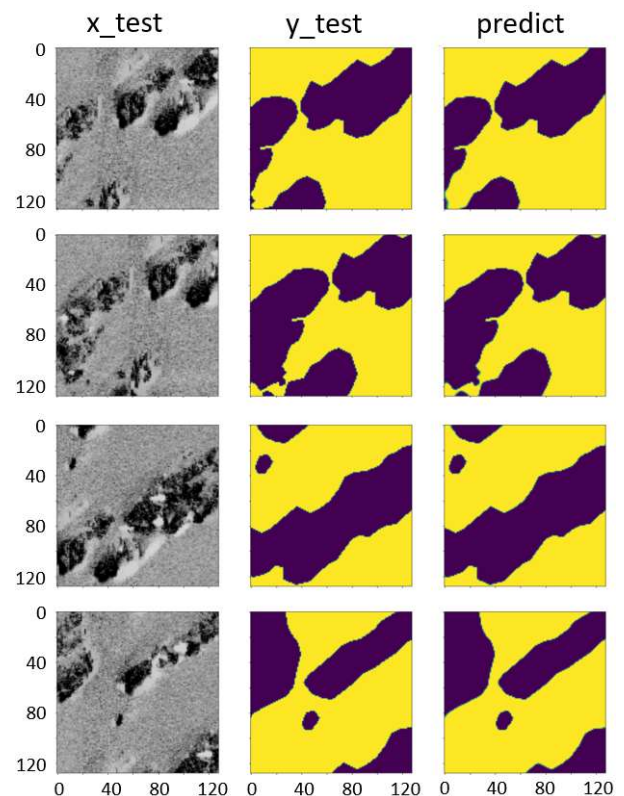

Figure 9 – Input data (x_test), output data (y_test) and model predictions.

Conclusion

The present work raised the level of automation of geohazard studies, implementing an intelligent system

through fully convolutional neural networks of the U-Net type in the segmentation of SONAR images. It achieved the ability to differentiate mud and carbonate features with an accuracy of 99.79%, proving the efficiency of such networks in the task of image segmentation.

Acknowledgments

References

LONG, J.; SHELHAMER, E.; DARRELL, T. 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p. 3431–3440.

NGI, 2012. Offshore Geohazards. Available in <https://www.ngi.no/eng/Projects/ICG-International-Centre-for-Geohazards/Offshore-Geohazards>. Access: June/2021.

RONNEBERGER, O.; FISCHER, P.; BROX, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. International Conference on Medical image computing and computer-assisted intervention. p. 234–241.

USGS. 2016. Side-scan sonar mosaic offshore of Petit Bois Island. Available in <https://www.usgs.gov/media/images/side-scan-SONAR-mosaic-offshore-petit-bois-island>. Accessed: June/21.

VARGAS, A. C. G.; PAES, A. V. C. N. 2016. Um Estudo sobre Redes Neurais Convolucionais e sua Aplicação em Detecção de Pedestres. P. 20-21.