Proceedings of the 14th International Symposium on Process Systems Engineering – PSE 2021+
June 19-23, 2022, Kyoto, Japan

Quantifying Subsea Gas Leakages using Machine Learning: a CFD-based study

Gustavo L.R. Caldas[a*], Thiago F.B. Bento[b], Roger M. Moreira[b] , Maurício B. de Souza Jr.[a]

[a] *Escola de Química, Universidade Federal do Rio de Janeiro, Rio de Janeiro 21941-909, Brazil*
[b] *Escola de Engenharia, Universidade Federal Fluminense, Niterói 24210-240, Brazil*
gustavo.caldas@eq.ufrj.br

Abstract

Subsea images captured on-site can be used to quantify gas leakage in the subsea environment. In this work, gas leakage in reduced conditions was simulated by Computational Fluid Dynamics (CFD). The aim is to develop a computational vision tool to quantify the leakage. The images generated from CFD simulations were processed by a convolutional neural network (CNN) structure, the U-Net. A class is attributed to each image pixel, and a post-processing algorithm computes the corresponding bubble area. Two cases were carried out: image segmentation into two (water and bubble) and three classes (bubble interface included). The multi-class U-Net shows a good agreement with CFD results compared to the binary one because separating the pixels into just two categories leads to bubble diameter overestimation. Hence, this method is of potential use in fault detection and diagnosis and could support the decision-making process on deepwater leakage remediation.

Keywords: gas leakage; machine learning; convolutional neural network; process safety.

1. Introduction

Subsea oil and gas activities demand safety procedures and constant monitoring to prevent impact on marine ecosystems and financial losses for the operating companies (Figueredo et al., 2022). Several resources might take hold for this purpose. For instance, real-time leakage filming is possible with the Remotely Operated Vehicles (ROV) equipped with a camera onboard. These images, however, provide information only on whether the leak is occurring. For a better assessment, it is of great interest to develop a quantitative tool to support the decision-making process of intervention.

A possible parameter for the leak estimation is the bubble diameter (Jamialahmadi et al., 2001), which could be computed using image processing techniques. More recently, convolutional neural networks (CNN) - a type of Machine Learning (ML) algorithm – became part of these techniques (Goodfellow et al., 2016). CNNs are sparsely connected neural networks, i.e., not all neurons are connected to the ones of the subsequent layers. As a result, it saves plenty of computational resources when dealing with tensor data such as images and sounds (Krizhevsky et al., 2012). In a CNN structure, the first argument is the input, and the second one, the kernel (filter). Typically, the input is a tensor containing the image height, width, and input channels (colors). The output is called the feature map, which stores the characteristics of the input data and simultaneously reduces its size by using a kernel smaller than the matrix – this is the reason for the sparse connectivity. The kernels' number, shape, and activation function are hyper-parameters defined by the user (Goodfellow et al., 2016).

Convolutional neural networks have already been applied to fault detection and diagnosis problems. Wu and Zhao (2018) verified its usefulness on the Tennessee Eastman process. The relation between different process variables and sampling time is concatenated into two-dimensional matrices, adequate for CNN computing. The fault diagnosis rate scored 88.2 %. Li et al. (2018) proposed a CNN to detect chemical leakage in hydrocarbon tanks based on image recognition. They obtained 85.82 % accuracy. Bai et al. (2021) developed a real-time classifier of gas dispersion state in a bubble column using a novel CNN architecture named BubbleNet. It differentiated flow conditions according to bubbles' size and shape after being trained to a labeled dataset. It scored 97.8 % and 97.5 % of the performance for the training and test, respectively.

In some chemical engineering applications with multi-phase flows, e.g., liquid-liquid extraction, it is fundamental to know the particle size distribution, a variable of interest for the transport phenomena control. Schäfer et al. (2019) investigated it using a particular convolutional neural network, the U-Net. This network was designed for image segmentation (Ronneberger et al., 2015), an application interested in localizing objects and boundaries by partitioning the image pixels into various segments. Thus, the U-Net permits phase fractions distinction. Another advantage is that post-processing enables the calculation of the droplet size distributions from the U-Net output. Therefore, the present study aims to develop a system capable of quantifying leakages in subsea processes employing the U-Net convolutional neural network.

2. Methodology

We carried out reduced model simulations of gas leakages employing Computational Fluid Dynamics (CFD). Reduced model is a technique that is used to save computational costs by downscaling the original phenomenon. For instance, it reproduces an event from the subsea scale to the laboratory. Gas leakages are released with different velocities (v) and from different orifice diameters (d). The initial value problem is solved via a finite volume method. The Volume of Fluid (VoF) method is employed to model the two-phase gas-liquid flow. Continuity and the unsteady RANS (Reynolds-Averaged Navier-Stokes) equations are satisfied in the fluid domain, with the classical κ-ε turbulence model being used. The CFD results are being validated with experiments and semi-empirical models. They agree on the trend found in the literature (Jamialahmadi et al., 2001). The simulation was carried out in ANSYS Fluent software, producing videos that represent the leakage. Each video frame generated an image set, totalizing 3159 images from the different conditions.

In a second step, the images are forwarded to a CNN model, called the U-Net structure. The main goal of this architecture is to classify each pixel individually as belonging to some class. The images are the input for training this network, and the targets are the masks created by a segmentation method. The CNN was developed in Keras environment employing Python with Tensorflow as backend. The segmentation was carried out using the unsupervised Otsu's methodology (Otsu, 1979) in the Scikit Image library written in Python. It is an algorithm whose aim is to find a threshold that can divide the pixels of a grayscale image into two clusters (classes): foreground f and background b. A threshold t is searched, such that the intra-class variance, represented in Eq. 1, is minimized (and the inter-class is maximized as well). The weights ω calculated contain the probabilities of a pixel to belong to one of the classes. In this case, classes are water (label zero) and bubble (label one). Given the importance of phase fraction when accounting bubble diameter, the problem was extended to multi-segmentation, in which the interface is labeled as number two. Multi-level thresholding can be performed as described by Otsu (1979).

$$\sigma_{\mathrm{w(t)}}^2 = \omega_b(\mathrm{t})\sigma_{\mathrm{b}}^2(\mathrm{t}) + \omega_f(\mathrm{t})\sigma_{\mathrm{f}}^2(\mathrm{t}) \tag{1}$$

The U-Net structure is shown in Figure 1. It is composed of a down-sampling part: successive blocks of convolutional 2D layers with filters of window dimension 3x3 and initialization “He” followed by 20 % dropout; a second convolutional layer; and a max pooling layer, which takes the maximum value over the window 2x2. In the next block, the number of filters is doubled (starting with 32). The second part comprises the up-sampling operations: transposed convolution (deconvolution) layers with filters 2x2 and stride 2x2. Information is concatenated from the corresponding feature maps of convolutional and deconvolutional layers. Another two convolutional layers are present on each block with half of the filters from the previous up-sampling block. The batch size is 128.

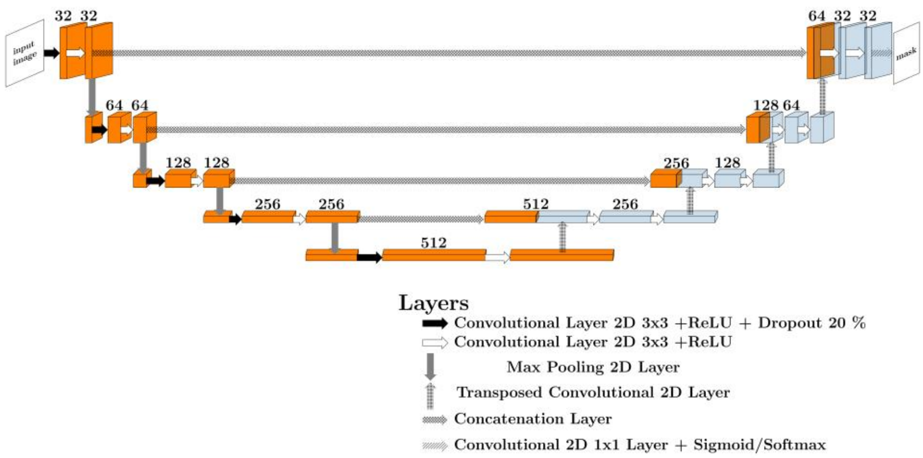


Figure 1: The U-Net architecture.

The total number of parameters for the binary class and the multi-label segmentation problems are 7,759,521 and 7,759,587, respectively. The metric used in this case was the Dice-Sørensen coefficient (Eq. 2a), which computes the similarity between the actual and predicted samples in relation to the group. It is important to use one-hot encoding format for the multi-class problem. Thus, the dice coefficient is extended for each class C (Eq. 2b). For the one-hot encoding format, the categorical cross-entropy (Eq. 3) was employed as a loss function to be minimized.

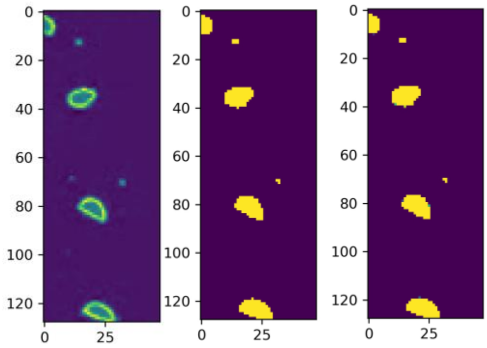
$$DSC = \sum_{\mathrm{n=1}}^{\mathrm{N}} \frac{\sum_{\mathrm{j=0}}^{128}\sum_{\mathrm{i=0}}^{48} \hat{y}_{\mathrm{ij,n}} \times \mathrm{y}_{\mathrm{ij,n}}}{\sum_{\mathrm{j=0}}^{128}\sum_{\mathrm{i=0}}^{48} \hat{y}_{\mathrm{ij,n}} + \sum_{\mathrm{j=0}}^{128}\sum_{\mathrm{i=0}}^{48} \mathrm{y}_{\mathrm{ij,n}}} \tag{2a}$$

$$DSC = \sum_{c=1}^{C} \frac{\mathrm{DSC_c}}{\mathrm{C}} \tag{2b}$$

$$CCE = -\frac{1}{N}\sum_{n=1}^{N}\sum_{j=0}^{128}\sum_{i=0}^{48}\sum_{c=1}^{C} y_{c,ij,n} \log \hat{y}_{c,ij,n} \tag{3}$$

3. Results and Discussion

Figure 2 shows an image sample (U-Net input), the corresponding binary segmentation mask, and the mask predicted. Bubble statistics are presented in Table 1.



(a) (b) (c)

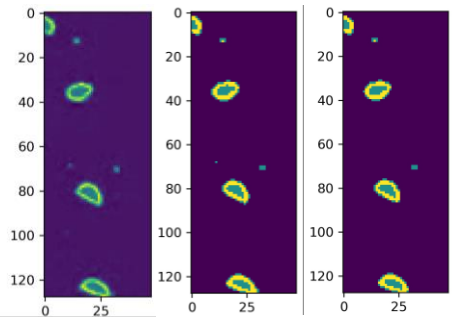
Figure 2: Snapshot of a sample: (a) U-Net input. (b) Mask generated by the binary Otsu's thresholding (U-Net target). (c) Mask predicted by the binary U-Net.

Table 1: Binary U-Net: predicted $\hat{d}_b$ against expected numerical diameter d_b.

d (mm)	v (m/s)	d_b (mm)	count	$\hat{d}_b$ (mm) mean	$\hat{d}_b$ (mm) std. dev.
0.5	0.25	6.45	350	7.82	0.36
0.5	0.625	7.22	369	9.79	1.16
0.5	1.0	7.69	374	10.19	1.95
1.0	0.24	6.37	317	9.44	0.94
1.0	0.37	6.46	311	9.95	1.72
1.0	0.5	7.28	377	11.74	2.01
5.0	0.02	6.70	373	8.67	0.37
5.0	0.055	8.05	308	10.51	0.96
5.0	0.09	8.11	380	12.14	2.30

The U-Net output is very similar to the target as the Dice-Sørensen coefficients for training and validation imply: 0.9915 and 0.9888, respectively. An overestimation is reported when comparing the expected numerical diameter d_b with the predicted one ($\hat{d}_b$). The reason is that the binary Otsu's thresholding does not set apart the interface and the bubble. This factor influences the area for calculation.

Due to the overestimation, it was decided to investigate further and add a phase fraction, turning the problem into a multi-class one. Figure 3 shows the analog result to Figure 2. The resulting mask resembles much more to the original image when compared to the previous case. The training was also successful. The multi-dice coefficient for training and validation reported 0.9507 and 0.9573, respectively. Similarly, the categorical cross-entropy loss found was 7.14 x 10^{-3} (training) and 5.34 x 10^{-3} (test). The predicted diameter by the multi-class U-Net shows a good agreement with the expected numerical diameter, as statistics shown in Table 2. Low standard deviations suggest that the biggest bubbles are relatively uniform for each dataset. Deviations from the actual values do not exceed 10 %, except for the 1.0 mm diameter crack cases.



(a) (b) (c)

Figure 3: Snapshot of a sample: (a) U-Net input. (b) Mask generated by the multi-label Otsu's thresholding (U-Net target). (c) Mask predicted by the multi-class U-Net.

Table 2: Multi-class U-Net: predicted $\hat{d}_b$ against expected numerical diameter d_b.

d (mm)	v (m/s)	d_b (mm)	count	$\hat{d}_b$ (mm) mean	$\hat{d}_b$ (mm) std. dev.
0.5	0.25	6.45	350	6.13	0.34
0.5	0.625	7.22	369	7.23	0.90
0.5	1.0	7.69	374	7.41	1.23
1.0	0.24	6.37	317	7.13	0.73
1.0	0.37	6.46	311	7.34	1.22
1.0	0.5	7.28	377	8.39	1.37
5.0	0.02	6.70	373	6.67	0.32
5.0	0.055	8.05	308	7.68	0.69
5.0	0.09	8.11	380	8.66	1.74

Regarding the model convergence, each model's total training time lasted around 1h30 min (2.5 - 3 min/epoch) in an Intel Core i5-10210. It has achieved less than 0.10 of loss in the fifth epoch, and after 15 epochs, more than 0.90 of Dice similarity coefficient.

4. Conclusions

A novel methodology was presented to quantify gas leakages that can be applied in a subsea environment, combining convolutional neural networks and a segmentation tool. The U-Net enabled the multi-segmentation post-processing to reach good predictability of the bubble diameter (less than 10 % deviation in general, the worst case was 15.24 % deviation). It is noteworthy that this performance was achieved with a relatively low amount of data (3159). For future works, it is suggested to expose the CNN to experimental data to validate the methodology.

Acknowledgments

[a]Financial support from the Human Resources Program of the National Agency of Petroleum, Natural Gas, and Biofuels – PRH-ANP, funded by oil companies qualified in the R&D and Innovation clause from ANP Resolution 50/2015 (process no. 041319).
[b]Financial support from PETROBRAS (process no. PETROBRAS/UFF/FEC 5900.0113248.19.9).

References

C. Bai, H. Park, Yong, C.Y. Ng, L. Wang, 2021, Classification of gas dispersion states via deep learning based on images obtained from a bubble sampler, Chemical Engineering Journal Advances, 5, 100064

A. K. M. Figueredo, M. B.de Souza, P. F. F. Frutuoso e Melo, C. A. Vaz Junior, J. Di Domenico, 2022, How does subsea processing safety policy impact safety? Lessons learned from the Brazilian experience, Journal of Loss Prevention in the Process Industries, 74, 104668

I. Goodfellow, Y. Bengio, A. Courville, 2016, Deep Learning, MIT Press, 775p.

M. Jamialahmadi, M.R. Zehtaban, H. Müller-Steinhagen, A. Sarrafi, J.M. Smith, 2001, Study of Bubble Formation Under Constant Flow Conditions, Chemical Engineering Research and Design, 79, 5, 523-532

A. Krizhevsky, I. Sutskever, G.E. Hinton, 2012, Image net classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 25, 1097–1105

L. Li, X. Jia, W. Tian, S. Sun, W. Cao, 2018, Convolution Neural Network based Chemical Leakage Identification, Computer Aided Chemical Engineering, 44, 2329 – 2334.

N. Otsu, 1979, A Threshold Selection Method from Gray-Level Histograms. In: IEEE Transactions on Systems, Man, and Cybernetics, 62, 2168-2909.

O. Ronneberger, P. Fischer, T. Brox, 2015, U-Net: Convolutional networks for biomedical image segmentation, In: Medical Image Computing and Computer-Assisted Intervention – Lecture Notes in Computer Science, 9351, 234–241, Springer

J. Schäfer, P. Schmitt, M.W. Hlawitschka, H. Bart, 2016, Measuring particle size distributions in multiphase flows using a convolutional neural network, Chemie Ingenieur Technik, 91, 11, 1688–1695

H. Wu, J. Zhao, 2018, Deep convolutional neural network model based chemical process fault diagnosis, Computers & Chemical Engineering, 115, 185-197