Convolutional Neural Networks for Underwater Pipeline Segmentation using Imperfect Datasets

Edgar Medina, Roberto Campos, José Gabriel R. C. Gomes, Mariane R. Petraglia, Antonio Petraglia
Electrical Engineering Program
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
Email: {emedina,roberto,gabriel,mariane,antonio}@pads.ufrj.br

***Abstract*—In this paper, we investigate a solution to the problem of underwater pipeline segmentation, based on an unbalanced dataset generated by a deterministic algorithm which employs computer vision techniques. We use manually selected masks to train two types of neural networks, U-Net and Deeplabv3+, to solve the same semantic segmentation task. We show that neural networks are able to learn from imperfect datasets, artificially generated by other algorithms. Deep convolutional architectures outperform the algorithm based on computer vision techniques. In order to find the best model, a comparison was made between the two architectures, thereby concluding that Deeplabv3+ achieves better results and features robust operation under adverse environmental conditions.**

***Index Terms*—Deep Learning, Convolutional Neural Networks, Semantic Segmentation.**

I. Introduction

Cameras are widely used in underwater pipeline inspection systems under good visibility conditions and pipeline proximity [1]–[4]. Some systems combine acoustic and visual information [5] or side scanning sonar, sub-bottom profiler and a magnetometer [6], to improve on robustness and result reliability.

Vision-based systems typically use algorithms such as Hough Transform [7] for pipeline detection, and Kalman [1] or particle filtering [3] for pipeline tracking. Most of these systems were developed for controlled environments, which renders them not applicable to real subsea scenarios.

The increasing use of deep learning in contemporary problems leads to most of the video inspection systems employing end-to-end deep neural networks to perform tasks such as automatic inspection, object detection, and image segmentation [8], [9]. Many deep architecture approaches to image segmentation have been proposed in recent literature, including spatial pyramid pooling [10]–[12], encoder-decoder [13]–[17], and encoder-decoder with atrous convolutions [18], [19]. The latter achieved the highest accuracy in many semantic segmentation online competitions [19].

Manual video annotation is a time-consuming, tedious and error-prone task. Some methods have been developed to alleviate such problems. In [20], generative neural networks for semi-supervised learning produced images and masks. Similar training procedures using synthetic data have been used over recent years [21], [22]. Other methodologies have adopted several alternative algorithms to generate masks [23], [24].

This paper investigates deep neural network techniques applied to the segmentation of underwater pipelines, which are trained using artificially annotated imperfect datasets. In the first stage of the proposed system, the images are processed and the corresponding annotation masks are obtained by using a conventional computer vision algorithm. In the second stage, the best annotations are manually selected and applied to train three convolutional neural network models. In order to investigate the impact of the use of imperfect datasets on the final segmentation results, experiments are conducted using different color domains, training strategies, topologies and backbone networks.

II. Vision-based Inspection System

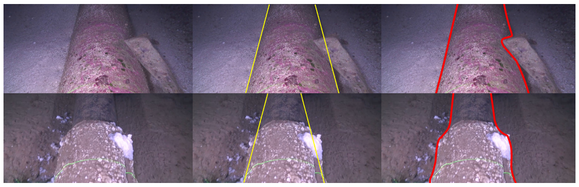
Videos acquired during real pipeline inspection missions were used in this work. Each video frame has 720×1280 color pixels. The overall raw data set, which was separated into 37 videos corresponding to an unbalanced variety of environments, has 28 GB. To reduce the computational cost in this task, a region of interest (ROI) was selected. This ROI removes textual information about the inspection presented on the top of each frame, then image resizing over the ROI was applied, thus resulting in images of 112×312 pixels that are used as neural network inputs. Figs. 1(a) and 2(a) show some inspection images after ROI selection and resizing.

A. Overview

The key aspect of our approach, which is much less costly in comparison to manual annotations (because the manual attention is now focused into choosing the best images from a set of conventional algorithm output images), is a procedure for training neural networks using artificially annotated imperfect data. The data set was generated by applying a deterministic algorithm based on computer vision and image processing techniques [25], in order to detect and track the pipeline position on all inspection videos. Since in this algorithm the pipeline is represented as a parallelogram, active contour model [26] was used to deform the pipeline edges to obtain a closer approximation to the real pipeline shape (some times deformed by the presence of algae, sand, etc.). Best results

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior Brasil (CAPES) - Finance Code 001, by CNPq and by FAPERJ.

were manually selected to be part of the data set. Fig. 1(b) shows the output borders from the algorithm reported in [25] and Fig. 1(c) corresponds to the deformed borders after using active contours. The area between the two borders is used as the pipeline mask.



(a) (b) (c)

Fig. 1: Intermediate results of the dataset composition: (a) input, (b) annotation using [25], (c) approximated borders for masks.

Then, for each frame and for each color channel (hue, saturation, and value), 32 bin background and 32 bin foreground histograms were concatenated, thus generating an environment descriptor with length equal to 192. Next, the DBSCAN algorithm [27], with $\epsilon = 8.0$ and minimum number of samples per group equal to 25, was used to find the approximate number of different environments present in the data set. This operation was followed by a visual inspection of the results, whose manual correction was performed to yield better environment separation. The final data set is composed of 69666 images and their corresponding masks.

B. Data structure

The data set created by the previously described method generates a set of environments, which are numbered from 1 to 31. Environment 32 was artificially generated with random background patches from the original images and split in 2671 and 8792 images for training and testing sets respectively. The database distribution with the number of samples per environment (Env) is detailed in Table I. We divided the number of environments in Table I so that roughly 50% of the data from Env 1 to Env 31 was assigned to the test set (indicated in bold in the table).

Unbalanced environment distribution was obtained, since several complex structures on the pipelines and significant background variations were found in particular cases. Because of significant intra-class background variations, the pipeline images corresponding to different environments must be handled carefully in order to ensure robust segmentation.

C. Neural Networks for Image Segmentation

We applied three different neural network architectures to address the pipeline segmentation problem. To select these architectures, we took into account the number of parameters and the accuracy reported in publicly available datasets.

U-Net: This architecture [14] was initially employed because it features fast convergence, and with the purpose of showing that our training procedure can generate enough labelled data to train a neural network even when data are not perfectly annotated. Some modifications were applied to the original architecture. Batch normalization was added after each convolutional layer. In contrast to the original U-Net model, the pooling kernel used in this topology had a size of (2,3) in the last downsampling and in the first upsampling (in this case using bilinear interpolation to perform the upsampling pooling). Finally, the output depth (number of channels) was equal to 1.

TABLE I: Number of images in the dataset per environment (Env). The Env numbers of the images assigned to the test set are indicated in bold.

Env	Images	Env	Images	Env	Images	Env	Images
1	333	9	120	17	299	**25**	438
2	4003	**10**	4230	18	438	**26**	38
3	393	11	28	**19**	57	**27**	461
4	8726	12	43	20	2066	**28**	48
5	347	13	15	21	255	**29**	14461
6	2738	14	1298	22	55	30	144
7	2032	15	6352	23	53	31	4931
8	3103	16	273	24	425	**32**	11463

SegNet: This architecture [17] performed deep fully convolutional neural network for semantic pixel-wise segmentation. It is composed of an encoder network and a corresponding decoder, followed by a final pixelwise classification layer. Each convolutional layer in the encoder is connected to its respective layer in the decoder section by a shortcut connection, which is extremely important in order to avoid gradient vanishing during training.

Deeplabv3+: This architecture [19] achieved state-of-the-art image segmentation performance. It is more complex than the U-Net architecture. As a backbone neural network, we used a pre-trained ResNet-101 [28] because it can be easily used from current frameworks [16]. Some modifications were applied to the backbone such as to adapt the input image to the network. Firstly, the initial 3×3 max pooling operation was removed, in order to avoid a large downsampling factor, as the original paper argued that such approach might be harmful for segmentation tasks. Secondly, the skip connection to attach the encoder to the decoder is taken from the Conv2_x of ResNet-101, and the same connection applies to all variations. Thirdly, the pooling layer connected after Conv3_x had its kernel size changed to (2,3). Finally, the last pooling operation and the fully-connected layers were removed, and the output layer depth was modified to the number of classes, i.e., it was set to 1. The pooling (2,3) was chosen to have a divisible image size on each hidden feature map to feed the following network layers.

III. Experiments

In this section, we discuss the training strategy and implementation details adopted for each architecture, and the effects of using imperfect annotations. In addition, we analyze

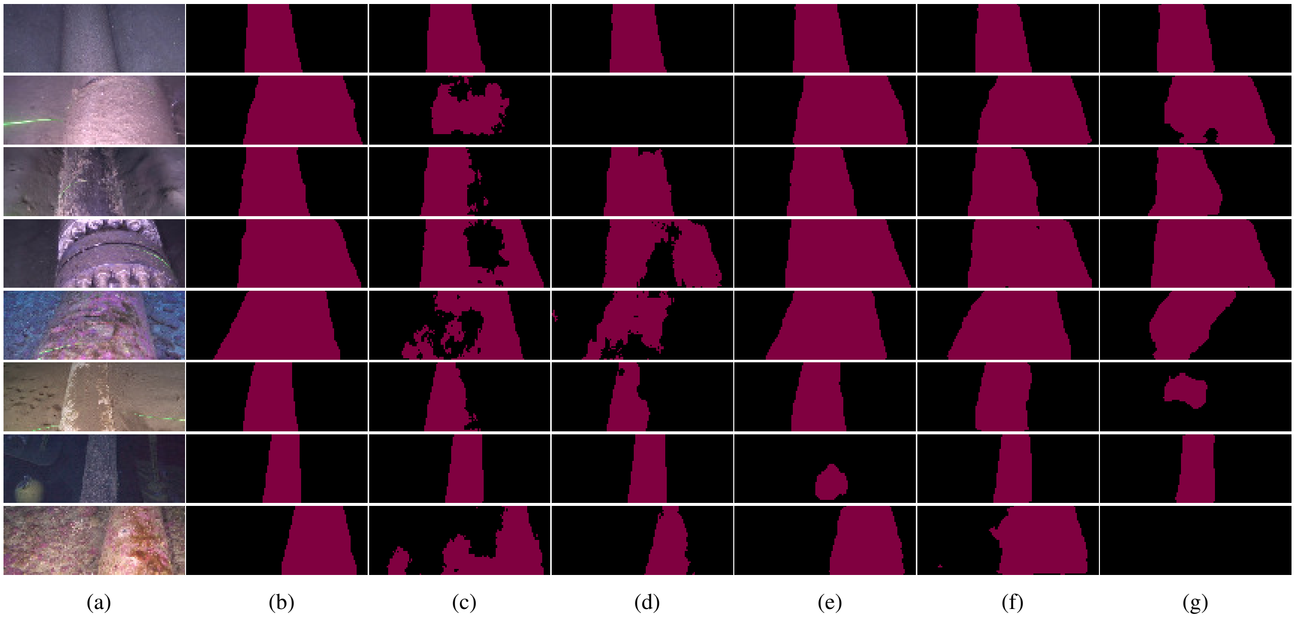


Fig. 2: Illustration of the results: (a) Input, (b) annotation, (c) U-Net (GRAY), (d) U-Net, (e) Deeplabv3+ with ResNet-101, (f) Deeplabv3+ with ResNet-101 (GRAY), and (g) Deeplabv3+ with ResNet-50.

and compare various networks and backbone variations of the Deeplabv3+ architecture.

A. *Implementation Details*

There are two problems to be solved before beginning to train a network: (i) define a loss function and (ii) determine a robust algorithm to obtain the model. To solve the first problem, the binary cross-entropy (BCE) and the soft-DICE coefficient were combined and adopted as loss function, while the mIOU (mean intersection over union) was tested as evaluation metric. We opted for the mIOU metric, which penalizes false positives more than false negatives, because the former are more critical for this application. The second problem to be solved was related to the avoidance of overfitting. For this purpose, regularizers were used in both networks, such as L2 norm, data augmentation and batch normalization.

The U-Net was trained using Adam optimizer, with a learning rate set to 1×10^{-3}, batch size equal to 16, and hyper-parameters β initialized with values 0.5 and 0.99, and decreased by a factor of 10 at epochs 15, 25 and 45 (out of a total of 50 epochs). On the other hand, the original Deeplabv3+ training setup was employed, with some minor changes. The initial learning rate was adjusted to 1×10^{-5} and the learning rate policy described in [29] was set to work only in the epochs 20, 40, 60, 80, 90 and 95 (from a total of 100 epochs). The Lagrangian factor that multiplied the L2 norm was 1×10^{-4} and the batch size was 16.

Data augmentation was applied to each image sample and to its respective mask, with the resulting database being composed of random rotations in the range from -15° to 15°, random translations up to 20% of the image size, random shear drawn having a uniform distribution in a range from -50° to 50°, and random horizontal and vertical flips generated by 0.5 and 0.2 occurrence probabilities, respectively.

B. *Results*

The U-Net, Segnet, and Deeplabv3+ having different Resnet backbone architectures were employed in this work, with the proposed methodology being evaluated in the test set described above. Quantitative results are shown in Table II. We applied two metrics: mIOU and frames per second (FPS). The FPS values were computed with all images previously stored in the GPU memory, i.e., without taking into account the communication time between the GPU and the main processor. A GPU GTX1080 was used for measurements of the FPS. Speed processing for [30] is not considered on Table II since the process is slower due to was entirely implemented on CPU.

TABLE II: Quantitative evaluation (mIOU and FPS) of object segmentation on the test set. Best values in bold.

Method	mIOU (%)	FPS
Computer vision [30]	96.30	-
U-Net (GRAY)	97.75	158.11
Segnet	96.58	**175.69**
Deeplabv3+(ResNet-18)	95.91	103.54
Deeplabv3+(ResNet-34)	97.05	72.57
Deeplabv3+(ResNet-50)	97.11	55.16
Deeplabv3+(ResNet-101)	**99.12**	31.46

We observed that the deep networks outperformed the computer vision algorithm for our data set (manually annotated for accuracy comparison). The Deeplabv3+ architecture using

ResNet-101 as network backbone produced the best mIOU from all models. However, the computational time required for achieving the output of this network was larger than the computational times of its smaller versions. In order to obtain the coefficients of the largest network we employed the ImageNet pre-trained weights and applied a fine-tuning on PASCAL VOC2012 before training it with our data set.

Although possibly the experimental procedure has a certain bias in the test set, since it is produced by the computer vision method [30], we compare a fine post-processed version of test images, as explained in section II-A, to minimize the bias. For this reason, the computer vision method reaches lower values than 100% in the metric. Some qualitative results comparing the performances of the six deep networks employed in this work, when applied to the test set, are shown in Fig. 2.

C. Add-on Study

In this section, we investigate the impact of the color domain, the loss function, and a variety of background images, on the the results. The motivation for conducting these experiments was the fact that underwater images are usually affected by the diversity of environments (explained above). The color domains used for this analysis were RGB, GRAY, nRGB, HSV and LAB.

In some scenarios, the training samples comprise the uncentered pipelines located far away from the camera or tight in the image. This can raise a bias issue, which is a typical problem of machine learning between the training and testing set. In order to avoid such bias in the network, we adopted a data augmentation strategy which adds a new environment composed only by background images (without foreground information). We called it Background Addition (BA). To evaluate this strategy, the U-Net was applied, whose results are detailed in Table III. In addition, we observed that the networks trained using GRAY images presented better results than the ones trained using RGB images, which might be due to the fact that color variations were removed and the topology was not totally robust to learn enough color and texture patterns.

An additional study on RGB images employing different loss functions showed that the use of DICE leads to suitable good results in mIOU, although during training some problems arose in cases when there were no visible pipeline. By contrast, training both networks using BCE improved mIOU in these cases, thereby decreasing the false-positive rate. This leads us to think that the sum of both loss functions may improve the mIOU for this application. The respective quantitative analysis is displayed in Table IV.

Finally, Table V shows that model performance improves on the testing set when the number of samples (composed by artificially generated masks) used for training increases. This experiment suggests that the imperfect annotations contributed to the improvement of the model.

D. Future Research Directions

In some complex cases where the pipeline is mostly occluded, the neural networks are not able to segment in correctly. For this reason, temporal information should be considered with the purpose of improving accuracy. Some methods such as LSTM addition [31] or prior information based on networks [32] show interesting results for the segmentation of video objects.

TABLE III: Comparison of mIOU results for different color domains. Best value in bold.

Color Domain	BA	mIOU
GRAY	✓	**97.75**
RGB	✓	97.12
LAB	✓	97.57
HSV	✓	96.62
GRAY		94.32
RGB		92.43
LAB		90.21
HSV		89.01
nRGB		81.98

TABLE IV: Comparison of mIOU results for different loss functions. Best value in bold.

Method	Loss	BA	mIOU
U-Net	BCE	✓	95.89
U-Net	DICE	✓	96.14
U-Net	DICE+BCE	✓	97.12
Deeplabv3+(ResNet-101)	DICE	✓	98.23
Deeplabv3+(ResNet-101)	DICE+BCE	✓	**99.12**

TABLE V: Comparison of mIOU results for different training set sizes.

Train Set(%)	mIOU
1	82.42
5	85.32
10	87.91
20	93.50
50	95.13
100	99.12

IV. Conclusions

In this paper, we addressed the problem of underwater pipeline segmentation, which is the first stage of an automatic vision-based inspection system. A training methodology for convolutional neural networks was introduced, which replaces manual annotations by cheaper and artificially generated datasets. This procedure showed that large architectures were satisfactorily trained from imperfect artificial masks, thereby surpassing the computer vision algorithm that generated this database. Comparisons with other networks, including topology variations, show that our proposed training strategy is robust and fast. Further studies on the impact of using different loss functions and color domains were carried out.

References

[1] A. Ortiz, M. Simó, and G. Oliver, "A vision system for an underwater cable tracker," *Machine Vision and Applications*, vol. 13, no. 3, pp. 129–140, jul 2002. [Online]. Available: https://doi.org/10.1007/s001380100065

[2] C. Cheng and B.-T. Jiang, "A robust visual servo scheme for underwater pipeline following," in *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2012, pp. 456–459.

[3] A. Ortiz, J. Antich, and G. Oliver, "A particle filter-based approach for tracking undersea narrow telecommunication cables," *Machine Vision and Applications*, vol. 22, no. 2, pp. 283–302, mar 2011. [Online]. Available: https://doi.org/10.1007/s00138-009-0199-6

[4] Y. Xu, Y. Zhang, H. Wang, and X. Liu, "Underwater image classification using deep convolutional neural networks and data augmentation," in *2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 2017, pp. 1–5.

[5] O. S. Goril M. Breivik Sigurd A. Fjerdingen, "Robust pipeline localization for an autonomous underwater vehicle using stereo vision and echo sounder data," pp. 7539 – 7539 – 12, 2010. [Online]. Available: https://doi.org/10.1117/12.839962

[6] W.-M. Tian, "Integrated method for the detection and location of underwater pipelines," *Applied Acoustics*, vol. 69, no. 5, pp. 387–398, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003682X07000916

[7] P. Drews Jr., V. Kuhn, and S. Gomes, "Tracking System for Underwater Inspection Using Computer Vision," in *2012 International Conference on Offshore and Marine Technology: Science and Innovation*. IEEE, 2012, pp. 27–30. [Online]. Available: http://ieeexplore.ieee.org/document/6257690/

[8] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494618302813

[9] K. S. Chahal and K. Dey, "A Survey of Modern Object Detection Literature using Deep Learning," *CoRR*, vol. abs/1808.0, aug 2018. [Online]. Available: http://arxiv.org/abs/1808.07256

[10] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, 2005, pp. 1458–1465 Vol. 2.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, sep 2015. [Online]. Available: http://arxiv.org/abs/1406.4729 http://ieeexplore.ieee.org/document/7005506/

[12] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," *Conference on Computer Vision and Pattern Recognition*, vol. abs/1612.0, dec 2016. [Online]. Available: http://arxiv.org/abs/1612.01105

[13] H. Noh, S. Hong, and B. Han, "Learning Deconvolution Network for Semantic Segmentation," in *2015 IEEE International Conference on Computer Vision (ICCV)*, vol. abs/1505.0. IEEE, dec 2015, pp. 1520–1528. [Online]. Available: http://arxiv.org/abs/1505.04366 http://ieeexplore.ieee.org/document/7410535/

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, may, pp. 234–241.

[15] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation," *Conference on Computer Vision and Pattern Recognition*, vol. abs/1611.0, nov 2016. [Online]. Available: http://arxiv.org/abs/1611.06612

[16] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, pp. 3309–3318. [Online]. Available: http://arxiv.org/abs/1611.08323 http://ieeexplore.ieee.org/document/8099836/

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[18] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," Tech. Rep., jun 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[19] L.-C. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," *Lecture Notes in Computer Science*, feb 2018. [Online]. Available: http://arxiv.org/abs/1802.02611

[20] N. Souly, C. Spampinato, and M. Shah, "Semi and Weakly Supervised Semantic Segmentation Using Generative Adversarial Network," *Conference on Computer Vision and Pattern Recognition*, mar 2017. [Online]. Available: http://arxiv.org/abs/1703.09695

[21] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting Unknown 3D Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Point Clouds," sep 2018. [Online]. Available: http://arxiv.org/abs/1809.05825

[22] W. Xu, Y. Li, and C. Lu, "SRDA: Generating Instance Segmentation Annotation Via Scanning, Reasoning And Domain Adaptation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6955–6963, jan 2018. [Online]. Available: http://arxiv.org/abs/1801.08839 https://ieeexplore.ieee.org/document/8578825/

[23] S. D. Jain and K. Grauman, "Active Image Segmentation Propagation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 2864–2873. [Online]. Available: http://ieeexplore.ieee.org/document/7780682/

[24] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao, "Learning from Weak and Noisy Labels for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 486–500, mar 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7450177/

[25] F. R. Petraglia, R. Campos, J. G. R. C. Gomes, and M. R. Petraglia, "Pipeline tracking and event classification for an automatic inspection vision system," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[26] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, jan 1988. [Online]. Available: https://doi.org/10.1007/BF00133570

[27] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, "A Density Based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases," *International Journal of Computer Applications*, vol. 3, no. 6, pp. 1–4, jun 2010. [Online]. Available: http://www.ijcaonline.org/volume3/number6/pxc3871038.pdf

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. abs/1512.0. IEEE, jun 2016, pp. 770–778. [Online]. Available: http://arxiv.org/abs/1512.03385 http://ieeexplore.ieee.org/document/7780459/

[29] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking Wider to See Better," *International Conference on Learning Representations*, vol. abs/1506.0, 2015. [Online]. Available: http://arxiv.org/abs/1506.04579

[30] R. E. Campos Ruiz, "Computer vision methods for underwater pipeline segmentation," Master's thesis, Federal University of Rio de Janeiro, 3 2018. [Online]. Available: www.pee.ufrj.br

[31] D. Nilsson and C. Sminchisescu, "Semantic Video Segmentation by Gated Recurrent Flow Propagation," *Conference on Computer Vision and Pattern Recognition*, vol. abs/1612.0, 2016. [Online]. Available: http://arxiv.org/abs/1612.08871

[32] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, "Efficient Video Object Segmentation via Network Modulation," *Conference on Computer Vision and Pattern Recognition*, vol. abs/1802.0, 2018. [Online]. Available: http://arxiv.org/abs/1802.01218