Novelty detection in underwater inspection videos

1st Evelyn C. S. Batista
Department of Electrical Engineering
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, Brasil
evelyn@puc-rio.br

2nd Wouter Caarls
Department of Electrical Engineering
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, Brasil
wouter@ele.puc-rio.br

3rd Leonardo A. Mendoza
Department of Electrical Engineering
Universidade do Estado do Rio de Janeiro
Rio de Janeiro, Brasil
leofome@hotmail.com

4rd Marco Aurélio Pacheco
Department of Electrical Engineering
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, Brasil
marco@ele.puc-rio.br

***Abstract*—This work consists of a study on the detection of anomalies in underwater inspection videos. Novelty detection is the identification of new or distinct data from a dataset, where the challenge is for an intelligent algorithm to be able to detect an input pattern as being previously unknown. Then, different models with different configurations were chosen, trained using different video frames and analyzed in order to find the best pattern detection model. The objective of this work is to contribute to the automatic detection of anomalies, which is currently a task carried out by specialists who analyze these videos. The model shows promise compared to other similar works in the area.**

***Index Terms*—Novelty Detection, Anomaly Detection, Autoencoder, Deep Learning.**

I. Introduction

Novelty detection or anomaly detection is a technique used in machine learning to identify patterns or instances that deviate significantly from the expected behavior or normal patterns within a dataset [1]. Projecting an anomaly detector is extremely complex due to the unpredictable aspect of the anomalies and their inaccessibility during former training. These factors reveal the unsupervised essence of the problem.

In the oil industry, subsea drilling represents a source of significant income in Brazil. Therefore, it is crucial to constantly monitor the equipment installed on the seabed, as any damage can result in serious environmental issues.

Currently, intelligent inspection systems are being widely used so that inspections are faster, more effective, and safer as they do not require a human in risky environments, especially underwater inspections [2] [3] [4] [5] [6] [7] [8]. In order for the inspection to acquire good results, it is necessary, in addition to analyzing recognized patterns, to also check for unknown patterns, called anomalies. It is important that the specialist very carefully analyzes the images that the UAV (Unmanned Aerial Vehicle), AUV (Autonomous Underwater Vehicle), or any other autonomous vehicle that is carrying out the inspection captures, to find any possible flaws or elements that should not be in the inspected environment. This procedure can take hours, even days, depending on the number of videos. An algorithm that automates this process can save the specialists a lot of time, making them focus on less tedious tasks.

In particular, underwater images usually have a very similar appearance due to low lighting and the uniformity of the ground. Sunlight decreases significantly as it penetrates the water, and most colors are absorbed at lower depths. As a result, underwater images often have a blue or greenish tone, can be turbid, and colors are generally less vibrant than on the surface [9]. In addition, the underwater ground is often uniform, with little variation in texture or color, which favors the training of the models.

These unique characteristics greatly hinder the identification of anomalies by experts. Therefore, this study employs novelty detection techniques to assist in this diagnosis.

Novelty detection is well-known in various fields, such as maintenance and monitoring [10] [11], process automation [12], video surveillance [13] [14], defect detection [15], supply chain [16], time series [17] and others.

As it can be observed, novelty detection finds application in various fields of research. In this study, its application is specifically focused on the oil industry, particularly in underwater inspections.

This highlights the significance of this work for the petroleum industry as a whole and for the Brazilian economy, as it is an important sector. This study will enable researchers to quickly and accurately identify anomalies in real-time during underwater inspections, saving specialists time.

As the demand for submarine resources continues to grow, the ability to monitor, detect, and correct anomalies effectively is of utmost importance. Therefore, companies in the oil and gas sector have been investing in research and development of

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001; The National Council for Scientific and Technological Development – CNPq under project number 314121/2021-8; and Fundação de Apoio a Pesquisa do Rio de Janeiro (FAPERJ) - APQ1 Program - E-26/010.001551/2019.

novelty detection solutions tailored to underwater conditions.

In this study, we focus on detecting anomalies in underwater inspection videos, testing different types of autoencoder networks which are known to be able to find anomalies in images [18]. Therefore, this work aims to train autoencoder models of distinct configurations using subsea inspection datasets and analyze which models provided better results.

II. THEORETICAL BACKGROUND

In this section, we briefly introduce the theory behind the concepts used in this paper: novelty detection, and autoencoder.

A. Novelty detection

Novelty detection requires something that stores events, such as memory or, more recently, representation in deep autoencoders. Following this, the capacity of remembering a certain sample is evaluated, measuring reconstruction errors. There are several applications in the anomaly detection area, such as video surveillance [19] [20] [21] [22], video inspections [23] [24] [25], medical images [26] [27] [28] [29] [30] and fraud detection [31] [32] [33] [34] [35].

B. Autoencoder

Autoencoders [36] are a neural network architecture class that aims to learn how to compress/reduce a datum (step known as an encoder) and then learn to reconstruct it from the version that was previously reduced (step known as a decoder). It is expected that the reconstructed data will suffer some (preferably minimal) information loss during the process (measured by the reconstruction loss). There are various types of autoencoder (see Figure 1), and in this work we use two of them: Dense Autoencoder (dAE) and Convolutional Autoencoder (cAE).

- **dAE:** is a naive autoencoder, containing dense and convolutional layers (Figure 1 - a);
- **cAE:** is an autoencoder network consisting only of convolutional layers (Figure 1 - b).

III. PROPOSED METHOD

Following this study's objective, many images were created to test and train the networks. Two datasets were created, one generated from images captured from the Airsim [37] simulator, which we call Synthetic dataset, and another dataset created from videos of ROV inspections, which we call Real dataset (Figure 2).

These two datasets were created to test the robustness of the model, both in simulated environments and in real life.

- Synthetic dataset: composed of 993 color images of 224x224;
- Real dataset: composed of 1475 color images of 640x480.

An autoencoder network is trained in order to detect anomalies. The proposal is to train the network with a dataset that contains only soil and pipeline images. After training, the network can be used for inference. As the autoencoder only learned characteristics from the training dataset images, if there is something different in the input image, the network decoder will not be able to recreate the different object (anomaly) in the image. Therefore, given that the output image has the same dimensions as the input, it is possible to subtract these images and generate a mask, respecting a threshold that will be chosen from the following sections (Figure 3). It is important to mention that the mask represents the pixels of the image anomaly.

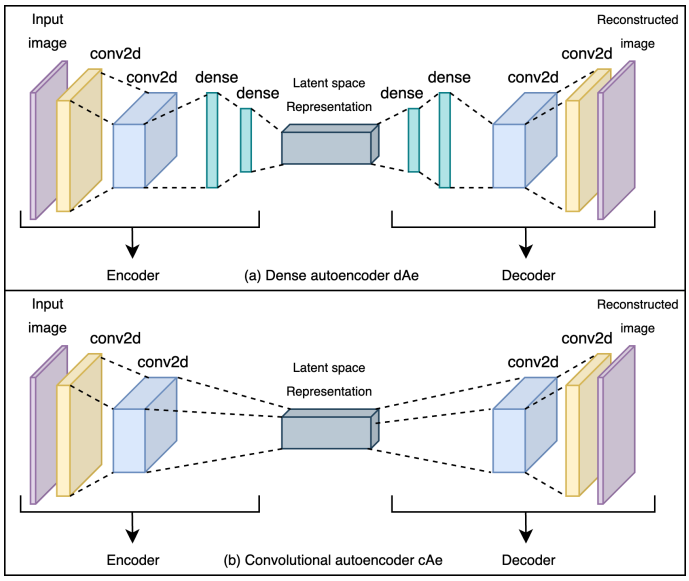

Fig. 1. An overview of different Autoencoder frameworks: Dense autoencoder (a) and Convolutional autoencoder (b).

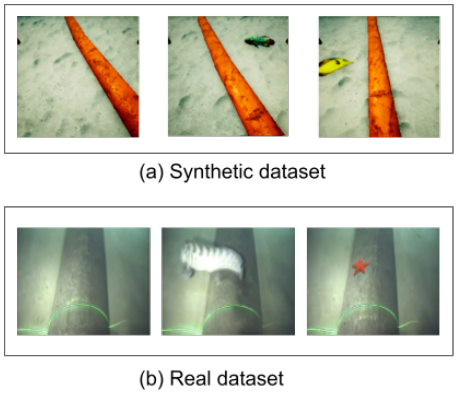

Fig. 2. Dataset images with synthetic(a) and real(b) image.

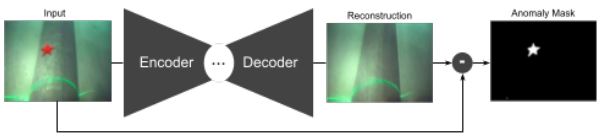

Fig. 3. The proposed novelty detection. A simple subtraction of the reconstructed image from the input reveals image anomalies.

TABLE I
ARCHITECTURE TABLE OF TRAINED MODELS WITH SYNTHETIC DATASET.

	dAE Model 1 Synthetic dataset	dAE Model 2 Synthetic dataset	cAE Model 1 Synthetic dataset	cAE Model 2 Synthetic dataset	cAE Model 3 Synthetic dataset
Encoder filters	32, 64	16, 32, 32, 64, 64	16, 8, 8, 8	16, 8, 8, 8, 8	16, 8, 8, 8, 4

TABLE II
ARCHITECTURE TABLE OF TRAINED MODELS WITH REAL DATASET.

	dAE Model 1 Real dataset	dAE Model 2 Real dataset	cAE Model 1 Real dataset	cAE Model 2 Real dataset	cAE Model 3 Real dataset
Encoder filters	32, 64	16, 16, 32, 32, 64, 64, 64	16, 8, 8, 8	16, 8, 8, 8, 8, 8, 8	16, 8, 8, 8, 8, 4, 4

A. The Architecture of the Network Chosen

To choose the best architecture for the model, several autoencoders with different depths were trained for the two datasets. Furthermore, for the dEA network, the latent layer size was varied (from 4 up to 1024), while the cAE networks were based on [39]. To evaluate the inference of each model, the Dice similarity coefficient [38](equation 1) was used.

$$\text{Dice Score} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{1}$$

Where X and Y represent two sets for which we want to measure the overlap. $|X|$ and $|Y|$ denote the cardinalities (i.e., the number of elements) of the sets X and Y, respectively. The intersection of the two sets, denoted as $X \cap Y$, represents the elements that are common to both sets.

The used encoder architectures are listed in Table I and II, where (the decoder is the opposite of the encoder, using *Conv2DTranspose*). All of them used *Conv2D* and *Conv2DTranspose* (functions from the Tensorflow library [40]) with stride 2 and 3×3 filters.

Network architectures for the real image dataset are deeper because these images are larger and more complex.

The hyperparameters used in the training of the both datasets are shown in Table III.

B. Training Summary

This work aimed to make a neural network recognize an novelty in a series of images. For this purpose, a neural network was trained. The network input, described above, requires at least one image per batch. The pre-processing only requires resizing each image to 224×224 (if Synthetic dataset is used) or to 640×480 (in case of Real dataset) and a normalization.

TABLE III
HYPERPARAMETERS AND VALUES

Hyperparameters	Values
Max epochs	200
Batch size	4
Learning rate	0.0001
Optimizer	Adam
Early stopping	min_delta=0.0001, patience=20

C. Choosing the Image Mask Threshold

After training the model, it is tested. The test consists of using an image (with or without novelty) as input to the model. As said before, the model will output an image of the same dimensions as the input, this output will be subtracted from the input image generating a mask (Figure 3). To better tune this mask, a threshold is chosen. So that the choice of this threshold is not random, tests were performed with thresholds 0, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 using the validation dataset. The threshold that maximizes the Dice score was chosen.

D. Choosing the loss function

The loss function, also known as the error function or cost function, is a fundamental part of deep learning models. It is responsible for quantifying the difference between the model's predictions and the actual values of the data set. The objective of the model is to minimize this error function by adjusting its parameters during the training process. There are several types of error functions, each suitable for different types of problems, such as classification or regression. Choosing the right error function can have a significant impact on the effectiveness of the model and how quickly it converges to an optimal solution.

In this study, SSIM (Structural Similarity Index Method) [42] and MSE (Mean squared error) were tested as a loss function. They are commonly used error functions in deep learning models.

The MSE measures the mean of the squares of the differences between the model's predictions and the actual values in the data set.

The SSIM Method is an image quality assessment technique widely used in image processing and computer vision. SSIM measures the structural similarity between two images, taking into account information about luminance, contrast and structure. The SSIM measure varies between -1 and 1, with values closer to 1 indicating greater similarity between the images. SSIM is especially useful for assessing the quality of images that have been compressed or otherwise distorted, as it takes human visual perception into account and can be more accurate than traditional measures such as MSE. SSIM is often used in applications involving image comparison, such as illegal copy detection, face recognition, and medical image monitoring.

For these reasons, MSE, SSIM and SSIM+MSE (using normalization to prevent metric overlap) were used as loss function, in order to obtain the best possible models.

E. Tests - Evaluation of results using Dice similarity coefficient

With the threshold chosen appropriately for each model, images (with and without anomalies in the same proportion)

TABLE IV
THRESHOLD FOR MODELS TRAINED WITH THE SYNTHETIC DATASET. (*) MODE OF THRESHOLDS THAT MAXIMIZE THE DICE SCORE.

	dAE Model 1 Synthetic dataset	dAE Model 2 Synthetic dataset	cAE Model 1 Synthetic dataset	cAE Model 2 Synthetic dataset	cAE Model 3 Synthetic dataset
Threshold	70(*)	80(*)	30	40	60

TABLE V
THRESHOLD FOR MODELS TRAINED WITH THE REAL DATASET. (*) MODE OF THRESHOLDS THAT MAXIMIZE THE DICE SCORE.

	dAE Model 1 Real dataset	dAE Model 2 Real dataset	cAE Model 1 Real dataset	cAE Model 2 Real dataset	cAE Model 3 Real dataset
Threshold	30(*)	30(*)	20	20	20

not used in training and in choosing the threshold are used for the test. A similar process to the one explained above is carried out. Images are the inputs of the trained model and its output is subtracted from the input image, generating a mask. Then the Dice score is calculated. The model with the highest Dice score is considered the best.

F. t-SNE Graphics Creation

In order to know how useful the representations created by the autoencoder networks are to separate the images with anomalies from the images without, the t-SNE algorithm [41] (t-Distributed Stochastic Neighbor Embedding, a dimensionality reduction technique that visualizes complex data while preserving non-linear relationships in a lower-dimensional space) is used. For this, a data sample from the test stage is used, it is input into the model and the encoder output is graphed using the t-SNE algorithm.

IV. RESULTS

The results of the training are reported below. The experiments were separated as follows:

- Best image mask threshold for each network architecture;
- Best loss function;
- Best model for Synthetic dataset;
- Best model for Real dataset.

A. Best image mask threshold for each network architecture

In this experiment, the objective is to choose a threshold that maximizes the Dice score, in order not to use a random threshold.

The threshold choice is highly important, as increasing the threshold could lead to the exclusion of crucial pixels for forming the mask that identifies the anomalous object. Unfortunately, this action would result in the loss of important information. To avoid this type of loss, we calculate the Dice score for each threshold in order to find the threshold that maximizes the Dice score.

The results of both the Synthetic (Table IV) and Real (Table V) datasets are found in the following tables.

TABLE VI
COMPARISON OF THE MEAN DICE SCORE OF MODELS TRAINED USING MSE, SSIM AND SSIM+MSE.

	dEA mean dice score	cEA mean dice score
MSE	0.4427	0.3712
SSIM	0.6314	0.5132
SSIM+MSE	0.8125	0.6510

TABLE VII
DICE-SCORE (MEAN AND STD. DEVIATION OF IMAGES) FOR TRAINED MODELS WITH SYNTHETIC DATASET. WHERE LD IS LATENT DIMENSION.

	dAE Model 1 (LD = 256) Threshold=70	dAE Model 2 (LD = 512) Threshold=70	cAE Model 1	cAE Model 2	cAE Model 3
Dice ($\mu \pm \sigma$)	0,90612 ± 0,12102	0,89964 ± 0,10237	0,82241 ± 0,15553	0,66944 ± 0,23307	0,55494 ± 0,19888

Analyzing the results, it is evident that the synthetic dataset showed a higher threshold compared to the real dataset. This can be attributed to the presence of less complex and more similar images in the synthetic set, resulting in masks with reduced levels of noise.

B. Best loss function

For all architectures created, the training was done using 3 variations of loss function: MSE, SSIM and SSIM+MSE, in order to discover the best one for the proposed task.

For both the dAE and cAE architectures, the loss SSIM+MSE presented the best results (Table VI), followed by the results of the loss being only SSIM and MSE.

Thus, we can conclude that when comparing the generated images with the original ones, it is more important to consider information about luminance, contrast and structure (calculated by SSIM) than just the difference in pixel values (calculated by MSE). However, the use of all information makes the result more accurate.

C. Best model for Synthetic dataset

After choosing the threshold for each model, they were tested. For the Synthetic dataset, the results are shown in Table VII.

Aiming to experiment with different networks of different depths, dAE and cAE networks were trained, and for dAE the latent dimensions were varied from 4 to 1024 resulting in 9 trained models for each dAE architecture variant, and from these variations the 3 loss functions were used. In other words, 27 models were trained for dAE models and 9 for cAE models.

For "dAE Model 1 Synthetic dataset" the highest score resulted from the use of the latent dimension of 256 and using the threshold selected in the previous step (Figure 4), while for "dAE Model 2 Synthetic dataset" the latent dimension that maximizes the score using the selected threshold is 512.

The best results came from the dAE models, using the SSIM+MSE loss. The best of these models, as seen in table 5, is the dAE Model 1.

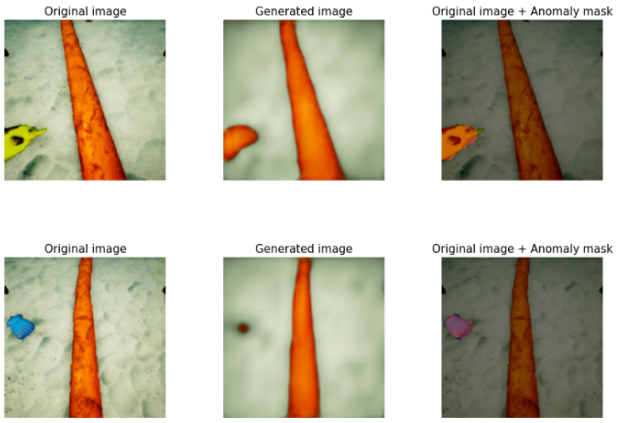

Fig. 4. Images resulting from "dAE Model 1 Synthetic dataset" inference

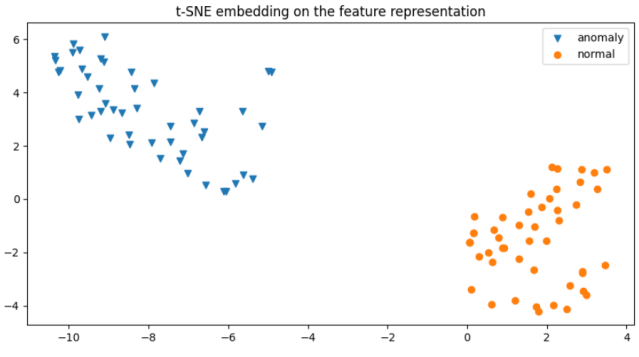

Fig. 5. t-SNE result of model "dAE Model 1 Synthetic dataset" encoder representation

It can be concluded that dense autoencoders work well for non-deep models that use images with few complexities. And the cAE models achieved moderate results due to the occurrence of noise in some cases, which generated inadequate masks.

The t-SNE was used in order to observe if the embeddings of data with anomalies can be separated from those without anomalies. The best possible result would be if the two classes were divided into well-defined clusters.

The t-SNE result, generated based on the best trained model and its embeddings (Figure 5), clearly illustrates the segregation between the representations of images with anomalies and those without. This highlights the effectiveness of the model in successfully performing its function of distinguishing between anomalous and non-anomalous images.

D. Best model for Real dataset

Using the real image dataset, experiments similar to the Synthetic dataset were performed. The results of the models were are shown in Table VIII.

For this dataset, the architecture using dAE with many convolutional layers (dAE Model 2) and with a latent dimension of 512 obtained the best result (Figure 6).

In this dataset, the worst results were obtained in the cAE models, the generated images were not very different from the original ones, however had a lot of noise, which did not occur in dAE models, which filtered out the existing noise due to the low latent space dimensionality.

TABLE VIII
DICE-SCORE (MEAN AND STD. DEVIATION OF IMAGES) FOR TRAINED MODELS WITH REAL DATASET. WHERE LD IS LATENT DIMENSION.

	dAE Model 1 (LD = 512) Threshold=60	dAE Model 2 (LD = 512) Threshold=60	cAE Model 1	cAE Model 2	cAE Model 3
Dice ($\mu \pm \sigma$)	0,76289 ± 0,24197	0,79379 ± 0,27118	0,54545 ± 0,14271	0,56118 ± 0,15140	0,53711 ± 0,15029

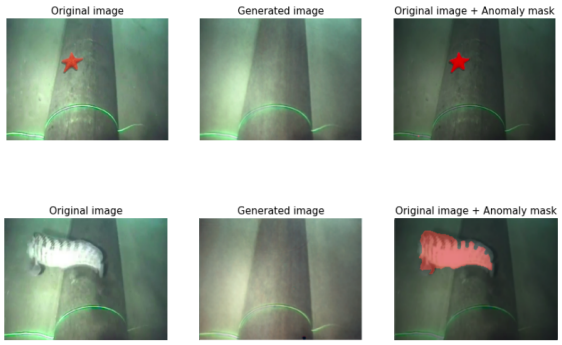

Fig. 6. Images resulting from "dAE Model 1 Real dataset" inference

In some of the results, some areas of the images were more segmented than necessary, but this can be advantageous, even if it may lead to an increase in the number of false positives - that is, regions that are mistakenly identified as belonging to the object of interest - it is preferable to have false positives above false negatives. This is because a false negative indicates that a region that should have been segmented as part of the object was mistakenly left out, causing the area not to be observed during the inspection. This can lead to inaccurate results and misinterpretations of the image. On the other hand, a false positive can be easily eliminated during the post-processing step.

Once again, to analyze the results, a graph was created using the embeddings of the most effective model, through the t-SNE technique (as illustrated in Figure 7). It can be inferred that even facing a dataset containing larger and more complex images, the model once again demonstrates its remarkable ability in distinguishing between images with and without anomalies. This performance is clearly visible in the segregation of the two classes of embeddings, with only a few distinct representations in proximity.

V. CONCLUSIONS

This work represents an advancement in the submarine oil industry, with the objective of automating processes that pose challenges to human perception.

Several autoencoder architectures have been trained to find anomalies in subsea inspection images, for specialists to more easily find anomalies in inspection videos.

The choice of using autoencoders was motivated by the fact that most works use these networks to detect anomalies, but

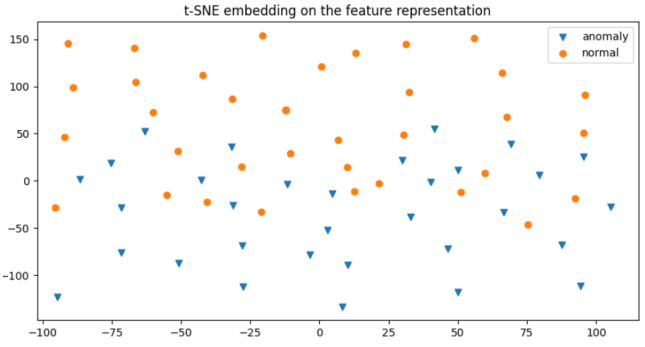

Fig. 7. t-SNE result of model "dAE Model 1 Real dataset" encoder representation

they use grayscale images, unlike this work, which also uses RGB images. In addition to featuring a characteristic blue or green tonality, these images generally exhibit a blurred appearance.

A study was carried out on the best architecture to be used to find anomalies in images of different sizes and origins, and it can be concluded that dense autoencoders and convolutional autoencoders are good networks to find anomalies in RGB images. However, the latter tends to generate noise that hinders the detection of anomalies.

Moreover, this work can evaluate the most suitable loss to be used in networks aiming to efficiently detect anomalies, reaching the conclusion that utilizing SSIM+MSE is the optimal choice for a loss function.

As future research, we will use Variational Adversarial Autoencoder (VAE) and Generative Adversarial Networks (GANs) to search for underwater image anomalies, in order to compare the results with those in this article.

REFERENCES

[1] Zimek, Arthur, and Erich Schubert. "Outlier detection, encyclopedia of database systems." Springer New York–2017. DOI 10 (2017): 978-1.

[2] Wu, Yinghao, et al. "Intelligent control method of underwater inspection robot in netcage." Aquaculture Research 53.5 (2022): 1928-1938.

[3] He, Jingyi, et al. "Multi-AUV Inspection for Process Monitoring of Underwater Oil Transportation." IEEE/CAA Journal of Automatica Sinica 10.3 (2023): 828-830.

[4] Waszak, Maryna, et al. "Semantic segmentation in underwater ship inspections: Benchmark and data set." IEEE Journal of Oceanic Engineering 48.2 (2022): 462-473.

[5] Amri, SQ Syamsul, AS Abdul Ghani, and MAS Kamarul Baharin. "Implementation of Underwater Image Enhancement for Corrosion Pipeline Inspection (UIECPI)." 2023 19th IEEE International Colloquium on Signal Processing & Its Applications (CSPA). IEEE, 2023.

[6] Jacobi, Marco. "Autonomous inspection of underwater structures." Robotics and Autonomous Systems 67 (2015): 80-86.

[7] Hou, Shitong, et al. "Underwater inspection of bridge substructures using sonar and deep convolutional network." Advanced Engineering Informatics 52 (2022): 101545.

[8] Stenius, Ivan, et al. "A system for autonomous seaweed farm inspection with an underwater robot." Sensors 22.13 (2022): 5064.

[9] Lee, Hoosang, et al. "Autonomous Underwater Vehicle Control for Fishnet Inspection in Turbid Water Environments." International Journal of Control, Automation and Systems 20.10 (2022): 3383-3392.

[10] Ahmad, Sabtain, et al. "Autoencoder-based condition monitoring and anomaly detection method for rotating machines." 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020.

[11] Russo, Stefania, et al. "Anomaly detection using deep autoencoders for in-situ wastewater systems monitoring data." arXiv preprint arXiv:2002.03843 (2020).

[12] Jeong, Sang-Ki, et al. "A study on anomaly detection of unmanned marine systems using machine learning." Measurement and Control 56.3-4 (2023): 470-480.

[13] Roka, Sanjay, and Manoj Diwakar. "Deep stacked denoising autoencoder for unsupervised anomaly detection in video surveillance." Journal of Electronic Imaging 32.3 (2023): 033015-033015.

[14] Chang, Yunpeng, et al. "Clustering driven deep autoencoder for video anomaly detection." Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. Springer International Publishing, 2020.

[15] Chow, Jun Kang, et al. "Anomaly detection of defects on concrete structures with the convolutional autoencoder." Advanced Engineering Informatics 45 (2020): 101105.

[16] Nguyen, H. Du, et al. "Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management." International Journal of Information Management 57 (2021): 102282.

[17] Gao, Honghao, et al. "Tsmae: a novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder." IEEE Transactions on network science and engineering (2022).

[18] Cui, Yajie, Zhaoxiang Liu, and Shiguo Lian. "A Survey on Unsupervised Anomaly Detection Algorithms for Industrial Images." IEEE Access (2023).

[19] Berroukham, Abdelhafid, et al. "Deep learning-based methods for anomaly detection in video surveillance: a review." Bulletin of Electrical Engineering and Informatics 12.1 (2023): 314-327.

[20] Patrikar, Devashree R., and Mayur Rajaram Parate. "Anomaly detection using edge computing in video surveillance system." International Journal of Multimedia Information Retrieval 11.2 (2022): 85-110.

[21] Sultani, Waqas, Chen Chen, and Mubarak Shah. "Real-world anomaly detection in surveillance videos." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[22] Zhu, Sijie, Chen Chen, and Waqas Sultani. "Video anomaly detection for smart surveillance." Computer Vision: A Reference Guide. Cham: Springer International Publishing, 2020. 1-8.

[23] Jin, Pu, et al. "Anomaly detection in aerial videos with transformers." IEEE Transactions on Geoscience and Remote Sensing 60 (2022): 1-13.

[24] Lu, Bingyu, Ding Xu, and Biqing Huang. "Deep-learning-based anomaly detection for lace defect inspection employing videos in production line." Advanced Engineering Informatics 51 (2022): 101471.

[25] Contreras-Cruz, Marco Antonio, et al. "Vision-based novelty detection using deep features and evolved novelty filters for specific robotic exploration and inspection tasks." Sensors 19.13 (2019): 2965.

[26] Tschuchnig, Maximilian E., and Michael Gadermayr. "Anomaly detection in medical imaging-a mini review." Data Science–Analytics and Applications: Proceedings of the 4th International Data Science Conference–iDSC2021. Wiesbaden: Springer Fachmedien Wiesbaden, 2022.

[27] Le, Khiem H., et al. "Learning from multiple expert annotators for enhancing anomaly detection in medical image analysis." IEEE Access 11 (2023): 14105-14114.

[28] Hansen, Stine, et al. "Anomaly detection-inspired few-shot medical image segmentation through self-supervision with supervoxels." Medical Image Analysis 78 (2022): 102385.

[29] Zhang, Haibo, et al. "Unsupervised deep anomaly detection for medical images using an improved adversarial autoencoder." Journal of Digital Imaging 35.2 (2022): 153-161.

[30] Tian, Yu, et al. "Unsupervised anomaly detection in medical images with a memory-augmented multi-level cross-attentional masked autoencoder." arXiv preprint arXiv:2203.11725 (2022).

[31] Alamri, Maram, and Mourad Ykhlef. "Survey of Credit Card Anomaly and Fraud Detection Using Sampling Techniques." Electronics 11.23 (2022): 4003.

[32] Jiang, Shanshan, et al. "Credit Card Fraud Detection Based on Unsupervised Attentional Anomaly Detection Network." Systems 11.6 (2023): 305.

[33] Hemdan, Ezz El-Din, and D. H. Manjaiah. "Anomaly Credit Card Fraud Detection Using Deep Learning." Deep Learning in Data Analytics: Recent Techniques, Practices and Applications (2022): 207-217.

[34] Pourhabibi, Tahereh, et al. "Fraud detection: A systematic literature review of graph-based anomaly detection approaches." Decision Support Systems 133 (2020): 113303.
[35] Ounacer, Soumaya, et al. "Anomaly Detection in Credit Card Transactions." Advanced Intelligent Systems for Sustainable Development (AI2SD'2019) Volume 4-Advanced Intelligent Systems for Applied Computing Sciences. Springer International Publishing, 2020.
[36] Bank, Dor, Noam Koenigstein, and Raja Giryes. "Autoencoders." arXiv preprint arXiv:2003.05991 (2020).
[37] Airsim (Accessed in: 2023-06-22). https://microsoft.github.io/airsim
[38] Gragera, Alonso, and Vorapong Suppakitpaisarn. "Semimetric properties of sørensen-dice and tversky indexes." International Workshop on Algorithms and Computation. Cham: Springer International Publishing, 2016.
[39] Boutarfass, Sanae, and Bernard Besserer. "Convolutional autoencoder for discriminating handwriting styles." 2019 8th European Workshop on Visual Information Processing (EUVIP). IEEE, 2019.
[40] Tensorflow (Accessed in: 2023-06-22). https://www.tensorflow.org
[41] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).
[42] Nilsson, Jim, and Tomas Akenine-Möller. "Understanding ssim." arXiv preprint arXiv:2006.13846 (2020).