Universitat
de les Illes Balears

DOCTORAL THESIS
2023

EFFICIENT IMPLEMENTATION OF DEEP NETS FOR VIDEO PROCESSING TO PRESERVE MARINE ECOSYSTEM SERVICES

Miguel Martín Abadal

Universitat
de les Illes Balears

DOCTORAL THESIS
2023

Doctoral Programme in Information and Communications Technology

EFFICIENT IMPLEMENTATION OF DEEP NETS FOR VIDEO PROCESSING TO PRESERVE MARINE ECOSYSTEM SERVICES

Miguel Martín Abadal

Thesis Supervisor: Yolanda González Cid
Thesis tutor: Yolanda González Cid

Doctor by the Universitat de les Illes Balears

Declaration of Authorship

I, Miguel MARTÍN ABADAL, declare that this thesis titled, "Efficient implementation of Deep Nets for video processing to preserve marine ecosystem services" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Supervisor's Agreement

I, Dr. Yolanda GONZÁLEZ CID, Ph.D. in Industrial Engineering and Professor at Department of Mathematics and Computer Science, University of the Balearic Islands, attest that:

this dissertation, titled Efficient implementation of Deep Nets for video processing to preserve marine ecosystem services, submitted by Miguel MARTÍN ABADAL for obtaining the degree of Doctor in Information and Communication Technologies, was carried out under my supervision and contains enough contributions to be considered a doctoral thesis.

Signed: _____

Date: _____

UNIVERSITY OF THE BALEARIC ISLANDS

Abstract

Higher Polytechnic School
Department of Mathematics and Computer Science

Doctor in Information and Communication Technologies

Efficient implementation of Deep Nets for video processing to preserve marine ecosystem services

by Miguel MARTÍN ABADAL

Marine ecosystems provide multiple services to humans, including provisioning services, such as seafood or fossil energy; regulating services, like coastal protection or water purification; cultural services, as tourism or spiritual benefits; and supporting services, like nutrient cycling or habitat provision.

The provided services are endangered by negative impacts that marine ecosystems are suffering due to multiple causes, some examples of which could be overfishing, habitat destruction, or plastic pollution. Therefore, there exists an urgency to develop new protective measures. One highlighted initiative is to develop scientifically and statistically robust monitoring methodologies and tools to control potential risks or assess the effectiveness of protective and recovery initiatives.

Ocean research and management is facing a new era, led by the technological developments in data collection, allowing the collection of vast amounts of data; and deep learning techniques, capable of processing the data and reducing its processing workload while increasing the spatial and temporal scope of conducted analysis. The marine science community is ready and willing to implement these new tools to a wide range of proposals towards the sustainability of marine ecosystems and its services.

The objective of this thesis is to study the applicability of deep learning solutions, along with computer vision, to develop new tools to preserve marine ecosystems and the offered services. Tools have been developed for three different tasks: *Posidonia oceanica* monitoring, jellyfish quantification and pipeline characterisation. In their development, diverse deep convolutional network model types and architectures have been trained and tested with data gathered from a variety of sources and under different environmental conditions. Additionally, the developed tools have been deployed into diverse platforms and adapted to its features and limitations.

These implementations cover a wide spectrum of scenarios where deep convolutional networks have been applied with good results, automating the data analysis process, expanding the temporal and spatial scope of the analysis or surveys, and improving the repeatability of experiments to detect evolution trends. Thus, validating the proposed methodology to implement deep convolutional networks for video processing to preserve marine ecosystem services.

UNIVERSIDAD DE LAS ISLAS BALEARES

Resumen

Escuela Politécnica Superior
Departamento de Matemáticas e Informática

Doctor en Tecnologías de la Información y las Comunicaciones

Efficient implementation of Deep Nets for video processing to preserve marine ecosystem services

por Miguel MARTÍN ABADAL

Los ecosistemas marinos ofrecen múltiples servicios a los seres humanos, incluyendo servicios de aprovisionamiento como la producción de comida o energía fósil, servicios de regulación como la protección costera o la depuración de aguas, servicios culturales como el turismo o beneficios espirituales y servicios de apoyo como la circulación de nutrientes o la provisión de hábitat.

Estos servicios se ven amenazados por los impactos negativos que están sufriendo los ecosistemas marinos debido a múltiples causas. Algunos ejemplos podrían ser la sobrepesca, la destrucción del hábitat o la contaminación por plásticos. Por lo tanto, existe la urgencia de desarrollar nuevas medidas de protección. Una iniciativa destacada es el desarrollo de metodologías y herramientas de monitoreo científica y estadísticamente sólidas para controlar los potenciales riesgos o evaluar la efectividad de iniciativas de protección y recuperación.

La investigación y gestión de los océanos se enfrenta a una nueva era, liderada por los avances tecnológicos en la obtención de datos, que permiten la recopilación de grandes cantidades de datos; y técnicas de aprendizaje profundo, capaces de procesar los datos y reducir el tiempo de procesamiento a la vez que aumentan el alcance espacial y temporal de los análisis realizados. La comunidad científica marina está lista y dispuesta a implementar estas nuevas herramientas en una amplia gama de propuestas para la sostenibilidad de los ecosistemas marinos y sus servicios.

El objetivo de esta tesis es estudiar la aplicabilidad de soluciones de aprendizaje profundo junto con visión artificial para desarrollar nuevas herramientas con el fin de preservar los ecosistemas marinos y los servicios ofrecidos. Se han desarrollado herramientas para tres tareas diferentes: la monitorización de *Posidonia oceanica*, la cuantificación de medusas y la caracterización de sistemas de tuberías. Durante su desarrollo, se han entrenado y probado diversos tipos de modelos y arquitecturas de redes convolucionales profundas con datos recopilados de una variedad de fuentes y en diferentes condiciones ambientales. Adicionalmente, las herramientas desarrolladas han sido desplegadas en diversas plataformas y adaptadas a sus características y limitaciones.

Estas implementaciones cubren un amplio espectro de escenarios en los que se han aplicado redes convolucionales profundas con buenos resultados, automatizando el proceso de análisis de datos, ampliando el alcance temporal y espacial de los análisis o inspecciones, y mejorando la repetibilidad de los experimentos para detectar tendencias de evolución. Por lo tanto, se ha validado la metodología propuesta para la implementación de redes convolucionales profundas para el análisis de datos en entornos marinos para la preservación de sus ecosistemas y servicios.

UNIVERSITAT DE LES ILLES BALEARS

Resum

Escola Politècnica Superior
Departament de Matemàtiques i Informàtica

Doctor en Tecnologies de la Informació i les Comunicacions

Efficient implementation of Deep Nets for video processing to preserve marine ecosystem services

per Miguel MARTÍN ABADAL

Els ecosistemes marins ofereixen múltiples serveis als humans, incloent serveis d'aprovisionament com la producció de menjar o energia fòssil, serveis de regulació com la protecció costanera o la depuració d'aigües, serveis culturals com el turisme o beneficis espirituals, i serveis de suport com la circulació de nutrients o la provisió d'hàbitat.

Aquests serveis es veuen amenaçats pels impactes negatius que estan patint els ecosistemes marins degut a múltiples causes, alguns exemples podrien ser la sobrepesca, la destrucció de l'hàbitat o la contaminació per plàstics. Així doncs, hi ha la urgència de desenvolupar noves mesures de protecció. Una iniciativa destacada és el desenvolupament de metodologies i eines de monitorització científica i estadísticament sòlides per controlar els riscos potencials o avaluar l'efectivitat d'iniciatives de protecció i recuperació.

La investigació i la gestió dels oceans s'enfronta a una nova era, liderada pels avenços tecnològics en l'obtenció de dades, permetent la recopilació de grans quantitats de dades; i tècniques d'aprenentatge profund, capaces de processar les dades i reduir el temps de processament alhora que augmenten l'abast espacial i temporal dels anàlisis realitzats. La comunitat científica marina està llesta i disposada a implementar aquestes noves eines en una àmplia gamma de propostes per a la sostenibilitat dels ecosistemes marins i els seus serveis.

L´objectiu d´aquesta tesi és estudiar l´aplicabilitat de solucions d´aprenentatge profund juntament amb visió artificial per desenvolupar noves eines per tal de preservar els ecosistemes marins i els serveis oferts. S'han desenvolupat eines per a tres tasques diferents: la monitorització de *Posidonia oceanica*, quantificació de meduses i caracterització de sistemes de canonades. Durant el desenvolupament s'han entrenat i provat diversos tipus de models i arquitectures de xarxes convolucionals profundes amb dades recopilades d'una varietat de fonts i en diferents condicions ambientals. Addicionalment, les eines desenvolupades han estat desplegades en diverses plataformes i adaptades a les seves característiques i limitacions.

Aquestes implementacions cobreixen un ampli espectre d'escenaris on s'han aplicat xarxes convolucionals profundes amb bons resultats, automatitzant el procés d'anàlisi de dades, ampliant l'abast temporal i espacial de les anàlisis o inspeccions i millorant la repetibilitat dels experiments per detectar tendències devolució. Per tant, s'ha validat la metodologia proposta per a la implementació de xarxes convolucionals profundes per a l'anàlisi de dades en entorns marins per preservar els seus ecosistemes i serveis.

Publications

Parts of this thesis have been published in international journals, conference proceedings or as book chapters. Here is a list of all authored or co-authored publications that present relationship with the work developed in this thesis, as well as other publications not directly related to the thesis, but relevant in research.

Additionally, quality indexes are provided for the compendium of journal articles featured in the thesis, to validate its relevance.

Related Publications

Journal Articles

- **Martin-Abadal, Miguel**, Eric Guerrero-Font, Francisco Bonin-Font, and Yolanda Gonzalez-Cid (2018). "Deep Semantic Segmentation in an AUV for Online Posidonia Oceanica Meadows Identification". In: *IEEE Access* 6, pp. 60956–60967. DOI: 10.1109/ACCESS.2018.2875412
  Quality index: JCR2018 *Computer science, information systems*, IF 4.098, Q1 (23/155)

- **Martin-Abadal, Miguel**, Manuel Piñar-Molina, Antoni Martorell-Torres, Gabriel Oliver-Codina, and Yolanda Gonzalez-Cid (2021b). "Underwater Pipe and Valve 3D Recognition Using Deep Learning Segmentation". In: *Journal of Marine Science and Engineering* 9.1. ISSN: 2077-1312. DOI: 10.3390/jmse9010005
  Quality index: JCR2021 *Engineering, marine*, IF 2.744, Q1 (4/16)

- **Martin-Abadal, Miguel**, Gabriel Oliver-Codina, and Yolanda Gonzalez-Cid (2022b). "Real-Time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks". In: *Sensors* 22.21. ISSN: 1424-8220. DOI: 10.3390/s22218141
  Quality index: JCR2021 *Engineering, electrical & electronic*, IF 3.847, Q2 (95/276)

- **Martin-Abadal, Miguel**, Ana Ruiz-Frau, Hilmar Hinz, and Yolanda Gonzalez-Cid (2020a). "Jellytoring: Real-Time Jellyfish Monitoring Based on Deep Learning Object Detection". In: *Sensors* 20.6. ISSN: 1424-8220. DOI: 10.3390/s20061708
  Quality index: JCR2020 *Engineering, electrical & electronic*, IF 3.735, Q2 (82/273)

- Ana Ruiz-Frau, **Martin-Abadal, Miguel**, Charlotte L. Jennings, Yolanda Gonzalez-Cid, and Hilmar Hinz (2022). "The potential of Jellytoring 2.0 smart tool as a global jellyfish monitoring platform". In: *Ecology and Evolution* 12.11. e9472 ECE-2022-04-00522.R2, e9472. DOI: https://doi.org/10.1002/ece3.9472

- Eric Guerrero-Font, Francisco Bonin-Font, **Miguel Martin-Abadal**, Yolanda Gonzalez-Cid, and Gabriel Oliver-Codina (2021a). "Sparse Gaussian process for online seagrass semantic mapping". In: *Expert Systems with Applications* 170, p. 114478. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.114478

Conference Proceedings

- **Martin-Abadal, Miguel**, Ivan Riutort-Ozcariz, Gabriel Oliver-Codina, and Yolanda Gonzalez-Cid (2019). "A deep learning solution for Posidonia oceanica seafloor habitat multiclass recognition". In: *OCEANS 2019 - Marseille*, pp. 1–7. DOI: 10.1109/OCEANSE.2019.8867304

- Yolanda Gonzalez-Cid, Francisco Bonin-Font, Eric Guerrrero Font, Antoni Martorell Torres, **Abadal, Miguel Martin**, Gabriel Oliver Codina, Hilmar Hinz, Laura Pereda Briones, and Fiona Tomas (2021). "Autonomous Marine Vehicles and CNN: Tech Tools for Posidonia Meadows Monitoring". In: *OCEANS 2021: San Diego – Porto*, pp. 1–8. DOI: 10.23919/OCEANS44145.2021.9705792

- Francisco Bonin-Font, **Abadal, Miguel Martin**, Eric Guerrrero Font, Antoni Martorell Torres, Bo Miquel Nordtfeldt, Julia Maez Crespo, Fiona Tomas, and Yolanda Gonzalez-Cid (2021). "AUVs for Control of Marine Alien Invasive Species". In: *OCEANS 2021: San Diego – Porto*, pp. 1–5. DOI: 10.23919/OCEANS44145.2021.9705915

Book Chapters

- **Martin-Abadal, Miguel**, Ana Ruiz-Frau, Hilmar Hinz, and Yolanda Gonzalez-Cid (2020b). "The Application of Deep Learning in Marine Sciences". In: *Deep Learning: Algorithms and Applications*. Ed. by Witold Pedrycz and Shyi-Ming Chen. Cham: Springer International Publishing, pp. 193–230. ISBN: 978-3-030-31760-7. DOI: 10.1007/978-3-030-31760-7_7

Unrelated Publications

Journal Articles

- Isabel Vidaurre-Gallart, Isabel Fernaud-Espinosa, Nicusor Cosmin-Toader, Lidia Talavera-Martínez, **Martin-Abadal, Miguel**, Ruth Benavides-Piccione, Yolanda Gonzalez-Cid, Luis Pastor, Javier DeFelipe, and Marcos García-Lorenzo (2022). "A Deep Learning-Based Workflow for Dendritic Spine Segmentation". In: *Frontiers in Neuroanatomy* 16. ISSN: 1662-5129. DOI: 10.3389/fnana.2022.817903

Conference Proceedings

- **Miguel Martin-Abadal**, Yolanda González Cid, Joan Roig-Nomura, Jose Jesus Manas, Attila Nagy, Tomas Salom, and Carlos Alonso (2019). "Cloud Type Distinction Based on CNN: An Aid for Short-Term Weather Forecast". In: *Artificial Intelligence Research and Development - Proceedings of the 22nd International Conference of the Catalan Association for Artificial Intelligence, CCIA 2019, Mallorca, Spain, 23-25 October 2019*. Ed. by Jordi Sabater-Mir, Vicenç Torra, Isabel Aguiló, and Manuel González Hidalgo. Vol. 319. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 152–159. DOI: 10.3233/FAIA190118

Acknowledgements

I would like to express my gratitude to my supervisor, Yolanda González Cid, for her guidance throughout this project. I would also like to thank the rest of the *Systems, Robotics & Vision* group; and my friends and family, who supported me and offered deep insight into the study.

Funding

The work reported in this thesis was supported by Servicio de Ocupación de las Islas Baleares (SOIB), the European Social Fund (ESF), Garantia Juvenil from the Ministerio de Empleo y Seguridad Social and by the Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contract DPI2017-86372-C3-3-R.

Contents

List of Acronyms

AP	**A**verage **P**recision
ASV	**A**utonomous **S**urface **V**ehicles
AUC	**A**rea **U**nder the **C**urve
AUV	**A**utonomous **U**nderwater **V**ehicles
CNN	**C**onvolutional **N**eural **N**etworks
DBSCAN	**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
DGCNN	**D**ynamic **G**raph **C**onvolutional **N**eural **N**etwork
DVL	**D**oppler **V**elocity **L**ogger
ESPs	**E**cosystem **S**ervice **P**roviders
FN	**F**alse **N**egatives
FP	**F**alse **P**ositives
IEA	**I**nformation **E**xtraction **A**lgorithm
IMU	**I**nertial **M**easurement **U**nit
IoU	**I**ntersection **o**ver **U**nion
IUA	**I**nformation **U**nification **A**lgorithm
LOS	**L**ine **O**f **S**ight
mAP	**m**ean **A**verage **P**recision
ML	**M**achine **L**earning
nms	**n**on-**m**axima **s**uppression
Po	**P**osidonia **o**ceanica
ROC	**R**eceiver **O**perating **C**haracteristic
ROS	**R**obot **O**perating **S**ystem
ROV	**R**emotely **O**perated **V**ehicles
SVM	**S**upport **V**ector **M**achines
TN	**T**rue **N**egatives
TP	**T**rue **P**ositives
USBL	**S**hort **B**aseline acoustic **L**ink
UVMS	**U**nderwater **V**ehicle **M**anipulator **S**ystems

Chapter 1

Introduction

This chapter first introduces the motivation and context behind this thesis. Next, its main objectives are presented. Finally, it summarises the remaining document structure.

1.1 Context

1.1.1 Ecosystem services

Ecosystem services are the direct and indirect contributions of the natural environment and ecosystems to human well-being. Understanding the interactions between ecological and social systems is a fundamental domain of ecology and is crucial for mapping and managing ecosystem services. This requires an understanding of how ecosystems contribute to human welfare. However, quantifying the management consequences on ecosystem functions and the resulting changes in the value of goods and services depends on the complex interactions between social-ecological systems (Norgaard, 2010).

The "Millennium Ecosystem Assessment" (Hassan et al., 2005) distinguishes between four types of ecosystem services:

- **Provisioning services:** These are material or energy outputs from ecosystems, including food, water, raw materials, and other resources.
- **Regulating services:** These are services that ecosystems provide by acting as regulators, such as regulating the quality of air and soil or controlling floods and diseases.
- **Cultural services:** These are non-material benefits obtained from being in contact with ecosystems, including aesthetic, spiritual, and psychological benefits.
- **Supporting services:** Closely related to regulating services, these services allow the ecosystems to continue providing the other services. They include nutrient cycling, primary production, soil formation, and habitat provision.

As previously mentioned, understanding ecosystem services is a complex task that requires a strong foundation in ecology, including an understanding of the principles and interactions of organisms and the environment Maurer, 2009. The scales at which these entities interact can vary widely, from microbes to landscapes and from milliseconds to millions of years. Furthermore, an ecosystem can provide multiple types of services; for example, the same forest may provide a habitat for organisms, or recreation opportunities and wood for humans. There also exist complex relationships and exchanges of energy and materials between different ecosystems (Bennett, Peterson, and Gordon, 2009).

A suggested research agenda (Kremen, 2005) for the study of ecosystem services includes the following steps:

- Identification of Ecosystem Service Providers (ESPs): species or populations that provide specific ecosystem services and the characterization of their functional roles and relationships.
- Determination of community structure aspects that influence how ESPs function in their natural landscape, such as compensatory responses that stabilize function and non-random extinction sequences that can erode it.
- Assessment of key environmental factors that influence the provision of services.
- Measurement of the spatial and temporal scales on which ESPs and their services operate.

Marine ecosystem services

Marine ecosystems are aquatic environments with high levels of dissolved salt, including deep-sea oceans, estuaries, and coastal marine ecosystems, each of which has unique physical and biological characteristics.

Marine ecosystems are defined by their unique biotic and abiotic components, which support each other for survival. Biotic factors include plants, animals, and microbes; important abiotic factors include the amount of sunlight in the ecosystem, the amount of oxygen, salt, and nutrients dissolved in the water, proximity to land, depth, and temperature.

Marine ecosystem services result from a wide variety of resources that marine ecosystems provide and that are consumed, used, or enjoyed by people (Buonocore et al., 2020; Barbier, 2017; Häyhä and Franzese, 2014). Marine ecosystems provide services of all the previously mentioned types. For example, they provide energy, food, coastal protection, carbon sequestration, and recreational opportunities. Table 1.1 showcases a wide variety of marine ecosystem services.

Provisioning	· Seafood from plants and animals · Renewable and fossil energy · Raw materials · Genetic material · Water
Regulating	· Coastal protection · Carbon sequestration · Climate regulation · Waste treatment · Water purification
Cultural	· Entertainment · Tourism · Aesthetic · Spiritual benefits · Habitat and species value · Cultural heritage
Supporting	· Nutrient cycling · Habitat provision for plants and animals · Gene pool protection

TABLE 1.1: Marine ecosystems services.

These services highly rely on the interplay between biotic and abiotic factors, depending on the physical, chemical, and biological processes that support marine ecosystems. Ecosystem processes include biomass production, organic matter transformation, nutrient cycling, and physical structuring (Strong et al., 2015).

During the last few decades, marine ecosystems have undergone drastic changes at different scales due to multiple anthropogenic causes, including overfishing, eutrophication, invasive alien species, habitat destruction, plastic pollution, and climate change (Ani and Robson, 2021; González-Ortegón and Moreno-Andrés, 2021; Antao et al., 2020; Küpper and Kamenos, 2018). These changes affect the previously mentioned ecosystem processes and thus the biotic and abiotic factors and provided services, affecting human well-being.

There is an urgent need to expand the range of protection for marine ecosystems. Some of the main agencies in the matter, such as the European Environmental Agency (EEA, 2020) or the International Seabed Authority (ISA, 2020), propose diverse measures for the preservation of water and marine environments:

- Progressively develop, implement, and review an adaptive, practical, and technically feasible regulatory framework, based on the best environmental practices, to protect marine ecosystems.
- Conduct assessments to support the implementation and development of regulatory measures.
- Ensure public access to environmental information and facilitate networking for better communication, coordination, and cooperation in terms of data reporting, management, and information sharing.
- Develop scientifically and statistically robust monitoring programs and methodologies to prevent, reduce, or control the potential risk of harmful activities and to assess the effectiveness of any protective or recovery initiatives.

1.1.2 Deep learning

Machine learning is a branch of artificial intelligence and computer science that focuses on the use of data and algorithms to imitate the way humans learn, gradually improving accuracy.

Machine learning powers many aspects of modern society, from web searches and content filtering on social networks to recommendations on e-commerce websites. It is also increasingly present in consumer products, such as televisions or smartphones.

Machine learning systems require the design of a feature extractor that transforms raw input data into an internal representation or feature vector from which a neural network can detect or classify patterns.

Deep learning is a sub-field of machine learning that differs primarily in the fact that deep-learning systems automatically extract features to perform tasks without human intervention from labelled or unlabelled raw data.

Deep learning and neural networks are accelerating progress in areas such as computer vision (Chai et al., 2021), natural language processing (Otter, Medina, and Kalita, 2021), speech recognition (Nassif et al., 2019) or robotics (Morales et al., 2021), among many others.

When it comes to machine or deep learning, there exist diverse categories of models depending on how the learning process is performed:

- **Supervised learning:** For the training procedure, the input is a known training data set with its corresponding labels. The model compares its output with the ground truth label and calculates the difference using a predefined loss function to modify the weights of the neural network. Applications of supervised learning include classification or regression problems.
- **Unsupervised learning:** The models can infer a function to describe previously unknown patterns or hidden structures from unlabelled data, clustering it based on the discovered features. Applications of unsupervised learning include clustering or association problems.
- **Semi-supervised learning:** The models combine a small amount of labelled data with a large amount of unlabelled data, performing weak supervision during training where labelled data acts as sanity checks. These models are able to produce better results than unsupervised learning models without the need of spending resources on labelling the entire dataset.
- **Reinforcement learning:** The models use raw unlabelled data to interact with the environment and are trained on a reward and punishment mechanism, rewarding correct moves and punishing wrong ones. The correctness of an output depends on previous states and outputs, allowing the determination of an ideal behaviour within a specific context to maximize the desired performance. The main applications for reinforcement learning are within complex and variant environments, such as self-driving cars or trading and finances.

Focusing on deep learning, there exists a variety of algorithms that are distinguished by the type of input data, network structure or data processing methods. Although there is no categorical correspondence between tasks to perform and algorithms to use, some algorithms are better suited to perform specific tasks due to their characteristics. Some of the most common deep learning algorithms include Multilayer Perceptrons, Convolutional Neural Networks, Recurrent Neural Networks, Generative Adversarial Networks, Restricted Boltzmann Machines or Autoencoders.

This thesis will focus on the use of Convolutional Neural Networks (CNNs), which are specifically designed for computer vision applications such as classifying images or identifying areas or objects of interest. CNNs applications are numerous and include medical image processing, scene recognition, document analysis, and face or emotion recognition.

CNNs consist of architectures with multiple layers of convolutions that use mask matrices to extract key features from the input data. CNN architectures can be divided into two parts. The first one consists of an encoder, built using multiple convolutional layers along with pooling layers to reduce the input dimensionality. In this section of the architecture, the initial layers produce feature maps containing low-level information such as edges, as the network deepens, it extracts higher-level concepts such as whole objects.

The second part of the network varies depending on the application. For image classification, where no spatial information is needed, the resulting feature maps from the encoder are mapped into a fixed-length vector using fully connected layers, proposing a confidence percentage for each possible class.

On the other hand, for tasks that use spatial information, like the identification of areas or objects of interest, a decoder is built using convolutional and upsampling layers. The low-resolution high-level information of the encoder is transformed into a high-resolution low-level information output. Additionally, skip connections are added, permitting the decoder to access the low-level information from the encoder in order to prevent information loss. Figure 1.1 showcases both CNN types of structures.

There is a wide variety of CNN architectures that can extract different types of information from an image. Following, the most common types of deep CNN architectures, their structures, and their uses are presented:

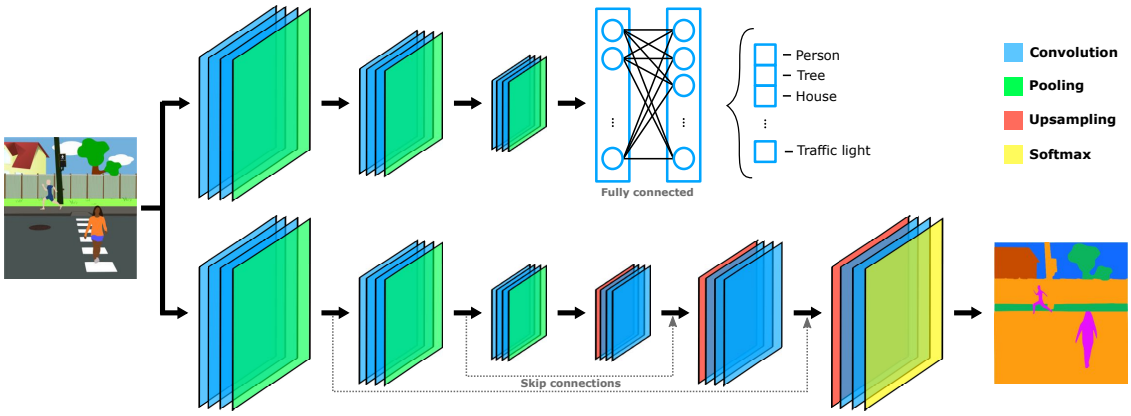


FIGURE 1.1: CNN types of structures. Top: fully connected structure. Bottom: Encoder-Decoder structure.

Image classification CNN architecture

Image classification is the task of categorising images into one or multiple predefined classes. Image classification CNN architectures use a fully connected structure (Figure 1.1) in which spatial information is lost, and a single label is assigned to an entire image. In these architectures, images can be processed quickly and often achieve results that surpass human-level accuracy (He et al., 2015). They are commonly used for simple classification tasks in medical imaging, satellite image processing, traffic control systems, and machine vision.

Object detection CNN architectures

Object detection is the task of identifying the presence of objects in an image and indicating their class and location with a bounding box. Object detection CNN architectures use an encoder-decoder structure (Figure 1.1), maintaining the spatial information needed to detect diverse objects and their position.

Deep learning object detection architectures can be divided into two types, depending on whether they use two-stage or one-stage algorithms (Lohia et al., 2021). Two-stage algorithms use a CNN network to extract image features, then, find possible candidate regions from the feature map using a region proposal network, and finally, perform sliding window operations on candidate regions to determine the object class and position (Girshick, 2015; Ren et al., 2015). One-stage algorithms use a single CNN that performs feature extraction, target classification, and position regression to directly predict the class and position of different targets. One-stage algorithms tend to have lower accuracy than two-stage algorithms but can process images much faster (Redmon et al., 2016; Liu et al., 2016). Object detection applications include autonomous driving, animal detection, medical feature detection, and surveillance.

Semantic segmentation CNN architectures

Semantic segmentation is the task of assigning a label to every pixel in an image, clustering the regions that belong to the same class. Semantic segmentation CNN architectures use an encoder-decoder structure (Figure 1.1) since spatial information is needed.

Deep learning semantic segmentation architectures can also be divided into two groups. Those with region-based algorithms, which use the same methodology of two-stage algorithms described in the Object detection architectures; and those with fully convolutional algorithms, using only a CNN to perform the segmentation task, equivalent to one-stage algorithms. Additionally, a combination of features from object detection and semantic segmentation architectures can be used to perform what is called instance segmentation, where every individual object in an image is detected, classified, and segmented (He et al., 2017; Zhang et al., 2020). Some applications for semantic and instance segmentation include autonomous driving, medical imaging, and document analysis.

Figure 1.2 illustrates the output differences when applying different CNNs architecture types to the same image.

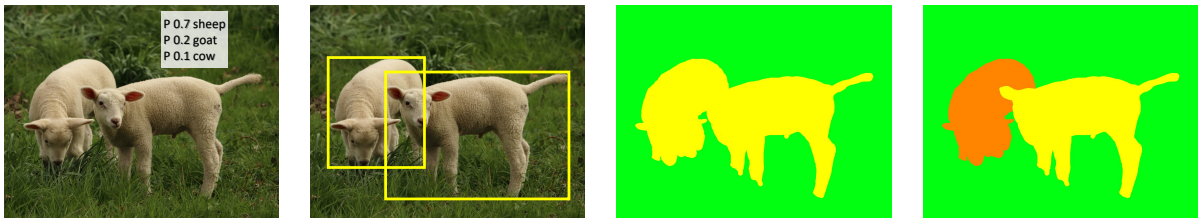


FIGURE 1.2: Output obtained when applying image classification (**a**), object detection (**b**), semantic segmentation (**c**) and instance segmentation (**d**) over the same image.

1.1.3 Deep learning implementation in marine ecosystems

As previously stated, marine ecosystems are diverse and provide multiple resources to the human population. However, anthropogenic factors are negatively impacting these ecosystems, endangering their balance and the services they provide. The scientific community aims to develop new techniques and mechanisms to provide reliable, up-to-date information on the state of marine ecosystems so that management decisions are well-informed.

In recent decades, technological developments in observation and data collection methods have been able to provide lots of information to ecologists. These developments include advances in visual cameras, echosounders, hydrophones, and environmental sensors such as temperature, current or salinity sensors. Concurrently, developments have taken place in the fields of data collection platforms, like underwater stationary observatories, floating buoys or marine vehicles such as Remotely Operated Vehicles (ROV), Autonomous Surface Vehicles (ASV) or Autonomous Underwater Vehicles (AUV). Figure 1.3 showcases different underwater data collection modalities.

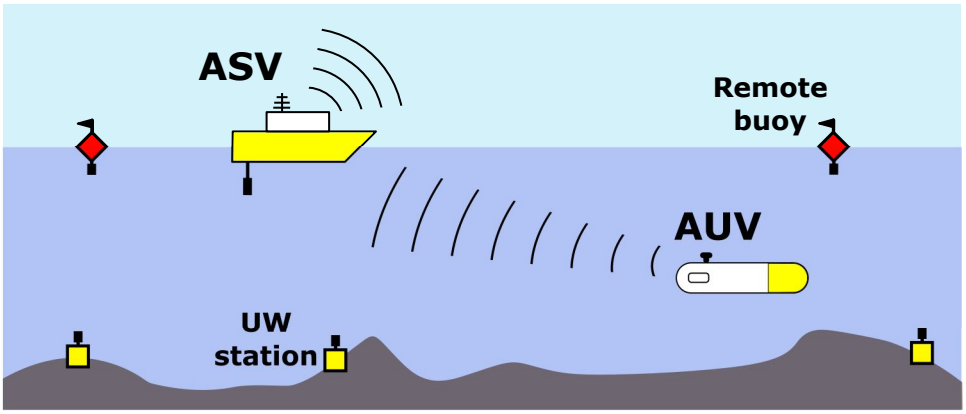


FIGURE 1.3: Examples of underwater data collection modalities.

The combination of these two factors has resulted in exponential growth of gathered information in both temporal and spatial terms, allowing researchers to better study underwater ecosystems and their biotic and abiotic factors (Bacheler et al., 2017). However, the curation and analysis of such vast amounts of data present some drawbacks if manual processing is needed, becoming a tedious and time-consuming task. The implementation of deep learning techniques allows to automate the data processing and reduce the time it takes, enabling the study of long temporal series or large areas and offering extra information to biologists.

Nonetheless, the application of deep learning implementations in marine ecosystems presents diverse challenges. Marine ecosystems are one of the less known ecosystems due to being hard to reach and operate on (Borja, 2014; St. John et al., 2016). Reasons for this include: insufficient oxygen, making it hard to perform manual labour as all procedures must be conducted by divers; high depth-increasing pressure, enforcing that all used data-gathering devices, exploration systems or any other equipment, must be able to function under these conditions; light transmission artefacts related to aquatic mediums that affect the quality of visual data such as light absorption, scattering or flickering; and the uncontrollable and rapid-changing nature of its environment, like variations in water turbidity or currents.

Therefore, despite the previously mentioned advances in data collection methods and platforms, the amount of data available from marine environments is much lower than others. There exist fewer public datasets, meaning that, in most cases, new datasets need to be generated, along with their ground truths. This implies that the size of datasets used to train and test the deep learning networks is usually relatively small, which is a factor to take into account when selecting a network architecture and designing its training.

An important aspect of any implementation is the ability to be executed in real-time, enabling the use of the generated information as input for other systems to make decisions on data collection processes, perform path replanning for exploration tasks, or take immediate action for protection tasks.

Additionally, as communication methods like cable or WiFi are typically unavailable in underwater scenarios, implementations benefit from being deployable and executable directly from the data collection platform without the need for information exchange. Implementations should be efficient and have low computational costs, considering the limited computational power and battery life of these platforms due to the constraints of working in underwater environments.

Monitoring biodiversity

Being able to process large temporal and spatial data is crucial for marine biodiversity monitoring. It allows to better study animal behaviour and early detect growing or declining trends in their numbers, as well as in algae coverage areas or any other important information that can be extracted through CNNs.

A great variety of CNN object detection architectures have been applied to count, measure or log the presence of multiple marine species such as fish and corals on underwater imagery (Li et al., 2015; Villon et al., 2016; Li and Du, 2022; Coro and Bjerregaard Walsh, 2021), whale echolocation clicks on spectrograms (Bermant et al., 2019); or plankton (Dai et al., 2016; Py, Hong, and Zhongzhi, 2016; Li et al., 2021) and algae (Park et al., 2022) on microscopic imagery, among others.

Semantic segmentation architectures are mainly used for extracting information of biodiversity from the benthic zone. In (Alonso et al., 2019) Alonso et al. make use of a semantic architecture along with sparsely labelled data to perform coral segmentation. Mohamed et al. in (Mohamed, Nadaoka, and Nakamura, 2022) use underwater imagery from a towed camera for automated segmentation of benthic habitats using unsupervised algorithms. Other works make use of CNNs to perform a patch-based classification of images containing seagrass meadows and generate a semantic segmentation after merging all patches (Gonzalez-Cid et al., 2017; Burguera, 2020). Finally, some applications make use of satellite imagery, for example, in (Gao et al., 2022), Gao et al. use a modified U-Net architecture to segment floating green algae from optical and SAR images.

Exploration and inspection

The development of ROVs and AUVs into the marine ecosystem has allowed access to deeper ocean regions, to examine larger areas and to operate on more complex underwater scenarios than what was possible with scuba divers. This, along with the usage of CNN to process the obtained information, offers a wide variety of possible implementations for exploration and inspection tasks.

Sonar imagery is widely used when performing exploration tasks, as it can quickly cover large areas while providing good enough resolution. Object detection architectures can be used to detect rather large objects like human bodies (Nguyen, Lee, and Lee, 2020) or warfare mines (Denos et al., 2017), or applied to fields such as archaeology, helping to identify shipwrecks of archaeological sites of interest (Nayak et al., 2021; Character et al., 2021). Semantic segmentation architectures can also be applied to sonar imagery, performing seafloor habitat mapping of a surveyed area and distinguishing the seafloor substratum (Burguera and Bonin-Font, 2020), or to discover new resource areas in deep-sea mineral exploration (Juliani and Juliani, 2021).

For inspection and manipulation tasks, the primary sensing modalities used are vision and laser, which can provide detailed information at short ranges. CNNs can also be applied to the information provided by these sensing modalities to perform underwater inspection and manipulation tasks in different scenarios like offshore oil and gas pipeline networks (Bharti, Lane, and Wang, 2020), metallic surfaces (Chen and Jahanshahi, 2018), or submarine communications cables (Thum et al., 2020).

Environment protection and surveillance

Satellite imagery, along with CNNs, can offer solutions for surveillance purposes such as boat detection to control illegal fishing, ballast water discharge, or anchoring (Kartal and Duman, 2019; Tang et al., 2020).

Segmentation networks can also be applied to satellite images to identify and segment oil spills (Huang et al., 2022; Yang, Singha, and Mayerle, 2022). In underwater imagery, object detection architectures can be used for ghost fishing gear recognition (Politikos et al., 2021) or underwater gas pipeline leak detection (Ahmad et al., 2022).

Additional information on the state of the art of the specific applications later presented in this document can be found in the publications included in their corresponding chapter.

1.2 Objectives

The main objective of this thesis is to develop deep learning-based tools using CNNs for image and video processing and efficiently implement them in real-world scenarios for the preservation of marine ecosystem services. It aims at improving current methods of gathering information by allowing the processing of data for longer periods of time, easing manual labour through the introduction of automatic systems, and increasing the accuracy of detection, annotation, and measuring tasks.

Specifically, this thesis aims to develop and implement tools for three tasks, listed below along with a description of the applications of the tool and specific objectives for each task.

1. ***Posidonia oceanica* monitoring**

Posidonia oceanica is an endemic plant of the Mediterranean sea that plays an important role in the marine and coastal ecosystems (Diaz-Almela and Duarte, 2008). Recent studies have shown a declining trend in its meadows extension (Marba and Duarte, 2010; Telesca et al., 2015). An important part of *Posidonia oceanica* control and recovery comes through monitoring and mapping its meadows, allowing for the early detection of decline trends or assessment of the effectiveness of recovery measures. The specific objectives for this application are:

   – Develop a tool able to automatically perform high-precision semantic segmentation of *Posidonia oceanica* meadows and their habitat in sea-floor images, to generate maps and monitor their status.

   – Online implementation into Robot Operating System (ROS) middleware (Quigley et al., 2009) to be deployed on AUVs or ASVs to serve as an input to a decision-time adaptive replanning algorithm to dynamically adapt the vehicle exploration path.

2. **Pipeline Characterisation**

There is an increasing need in performing underwater tasks like inspection and intervention on offshore oil and gas rigs or underwater pipeline networks (Yu et al., 2017; Jacobi and Karimanzira, 2013). This has motivated the development of AUVs equipped with sensors and manipulators, allowing to reach deeper and more complex underwater scenarios while reducing the associated risks of such tasks (Ridao et al., 2015; Heshmati-Alamdari et al., 2018). The specific objectives for this application are:

   – Design a system able to automatically identify and gather information from valves, pipes, and structural elements on underwater pipeline networks and position them in a 3D space.

   – Online implementation into *ROS* middleware to be deployed on AUVs or ASVs providing real-time information for inspection and manipulation tasks.

3. **Jellyfish detection and quantification**

Jellyfish have been recognised as an important part of marine ecosystems, providing multiple benefits to them (Hays, Doyle, and Houghton, 2018; Lamb et al., 2019). Recently, an increase in their numbers has been linked to global change and anthropomorphic causes (Richardson et al., 2009; Brotz et al., 2012), impacting human well-being (Lee et al., 2006; Purcell, Baxter, and Fuentes, 2013; Fenner, Lippmann, and Gershwin, 2010). Jellyfish monitoring efforts are often limited in terms of spatial and temporal coverage, resulting in uncertainty over the species population growth (Pitt et al., 2018). The specific objectives for this application are:

   – Develop a tool capable of automatically detecting different species of jellyfish and quantifying their presence over long periods of time.

– Implement the system online to deploy it into a network of buoys to generate real-time logs of jellyfish presence in a studied area.

Another goal of this thesis is to design, test, and validate a methodology for the development and efficient implementation of the previously mentioned tools. The proposed methodology is as follows:

1. Communicate with marine biologists and experts to better understand the problem and discuss possible solutions.
2. Study solutions from a technical viewpoint, accounting for the type of CNN, execution time, accuracy constraints, deployment platforms, etc.
3. Design efficient data collection experiments, marine environments are hard to reach and images can be affected by light transmission artefacts or environmental factors.
4. Collect rich and diverse data to train and test the CNNs on a wide variety of scenarios and environmental conditions.
5. Train and test the selected CNN, fine-tuning its hyperparameters taking into account the study stated in step 2.
6. Develop any post-processing code or algorithms needed to process the network output into useful information.
7. Efficiently implement the developed tool into deploying platforms, taking into account important factors such as computational power, heat dissipation, storage space, and communication networks.
8. Perform tests in real-world scenarios to ensure the tool's applicability and functionality.
9. Provide the necessary software and training to marine biologists and experts so that they can understand and use the developed tools.

1.3 Document Overview

The remainder of this dissertation is organised as follows:

Chapter 2 presents, through the journal article *"Deep Semantic Segmentation in an AUV for Online Posidonia oceanica Meadows Identification"* and conference article *"A deep learning solution for Posidonia oceanica seafloor habitat multiclass recognition"*, the work carried out on *Posidonia oceanica* monitoring, showcasing a deep learning based approach to automatically perform a high-precision semantic segmentation of *Posidonia oceanica* meadows and their habitat.

Chapter 3 covers, through the journal articles *"Underwater Pipe and Valve 3D Recognition Using Deep Learning Segmentation"* and *"Real-time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks"*, the work carried out on pipe and valve recognition and characterisation, detailing a system based on deep learning that automatically identifies and gathers 3D information from underwater pipeline networks for inspection and manipulation tasks.

Chapter 4 presents, though the journal article *"Jellytoring: Real-Time Jellyfish Monitoring Based on Deep Learning Object Detection"*, the work carried out on jellyfish detection and quantification, showcasing a deep learning tool to automatically log the presence of different species of jellyfish over a video feed.

Chapter 5 highlights the main contributions and discusses the relevance of the research. Finally, proposes diverse possible future lines of research.

Chapter 2

Posidonia oceanica monitoring

This chapter presents the work carried out on *Posidonia oceanica* monitoring.

Posidonia oceanica is an endemic plant of the Mediterranean sea which offers multiple benefits to the marine and coastal ecosystems (Diaz-Almela and Duarte, 2008). Recent studies evidence a significant decline of its meadows on a global scale (Marba and Duarte, 2010; Telesca et al., 2015). An important part of *Posidonia oceanica* control and recovery comes through monitoring and mapping of its meadows and the seafloor habitat where it develops, allowing for early detection of decline trends or assessment of the effectiveness of recovery measures. Currently, these monitoring tasks are mostly carried out by divers (Pizarro et al., 2017), making them slow and costly (Caughlan, 2001; Del Vecchio et al., 2018).

The objective of this work is to automatically perform a high-precision semantic segmentation of *Posidonia oceanica* meadows and their habitat in sea-floor images using deep learning techniques.

The first step was to collect the data to train and test the deep learning architecture. To do so, several hundred images of the seafloor containing *Posidonia oceanica* meadows under different conditions and sediments were collected using an AUV equipped with cameras. Next, a CNN semantic segmentation architecture was implemented and trained several times to obtain the best performing hyperparameters, distinguishing between *Posidonia oceanica* and background. Later, the selected CNN architecture was modified to perform multi-class segmentation, allowing the differentiation of other seafloor substrates such as sand, rocks, *Posidonia oceanica* matte or dead shoots.

The work carried out in this thesis regarding *Posidonia oceanica* habitat recognition is described in detail in two publications. The first one is a journal article explaining the data collection and dataset generation, the semantic segmentation network selection, hyperparameter tuning, validation, and online implementation. The second one is a conference article that presents the multi-class segmentation and validation.

Title: Deep Semantic Segmentation in an AUV for Online Posidonia oceanica Meadows Identification
Authors: **M. Martin-Abadal**, E. Guerrero-Font, F. Bonin-Font and Y. Gonzalez-Cid
Journal: IEEE Access
Published: 11 October 2018
Quality index: JCR2018 *Computer science, information systems*, IF 4.098, Q1 (23/155)

Title: A deep learning solution for Posidonia oceanica seafloor habitat multiclass recognition
Authors: **M. Martin-Abadal**, I. Riutort-Ozcariz, G. Oliver-Codina and Y. Gonzalez-Cid
Congress: IEEE Oceans
Date: 17-20 June 2019
Quality index: GGS Conference rating - B

Deep Semantic Segmentation in an AUV for Online *Posidonia oceanica* Meadows identification

Miguel Martin-Abadal*, Francisco Bonin-Font and Yolanda Gonzalez-Cid

Department of Mathematics and Computer Science. University of the Balearic Islands, 07122, Palma, Spain

ARTICLE INFO

The work presented in this preprint has been published in the journal *IEEE Access* as:

M. Martin-Abadal, E. Guerrero-Font, F. Bonin-Font and Y. Gonzalez-Cid, *Deep Semantic Segmentation in an AUV for Online Posidonia oceanica Meadows Identification*, IEEE Access, 2018, 6, 60956-60967.
DOI: 10.1109/ACCESS.2018.2875412

ABSTRACT

Recent studies have shown evidence of a significant decline of the *Posidonia oceanica* meadows on a global scale. The monitoring and mapping of these meadows are fundamental tools for measuring their status. We present an approach based on a deep neural network to automatically perform a high-precision semantic segmentation of the *Posidonia oceanica* meadows in sea-floor images, offering several improvements over the state-of-the-art techniques. Our network demonstrates outstanding performance over diverse test sets, reaching a *precision* of 96.57% and an *accuracy* of 96.81%, surpassing the reliability of labeling the images manually. Moreover, the network is implemented in an autonomous underwater vehicle, performing an online *Posidonia oceanica* segmentation, which will be used to generate real-time semantic coverage maps.

1. Introduction

Posidonia oceanica (P.o.) is an endemic seagrass species of the Mediterranean waters that forms dense and extensive meadows, offering many benefits to the marine and coastal ecosystems [7]. Recent studies have shown evidence of a decline at alarming rates of P.o. meadows on a global scale [16, 32]. For these reasons, the European Commission directive 92/43/CEE identifies P.o. as a priority natural habitat.

A very important part of P.o. control and recovery comes through monitoring and mapping of its meadows. These are fundamental tools for measuring their status, helping to detect decline trends early on, or address the effectiveness of any protective or recovery initiative.

Nowadays, monitoring tasks are mainly carried out by divers, who measure manually meadows descriptors such as extension, shoot density or lower limit depth [25]. Nevertheless, these processes tend to be slow, imprecise and very resource-consuming.

Other approaches to monitor P.o. include the use of: multi-spectral satellite imagery [26], acoustic bathymetry [19] or *Autonomous Underwater Vehicles* (AUV) equipped with different sensors, to extract information of P.o. meadows [23, 33]. However, these techniques suffer from lack of effectiveness in deep areas, in segregating P.o. from other algae types or are not able to perform a fully autonomous detection.

Recently, Burguera et al. [3] have achieved a fully autonomous detection by means of combining traditional image descriptors alongside *Machine Learning* (ML) using *Support Vector Machines* (SVM). Also, Gonzalez et al. [11] have explored the idea of using *Convolutional Neural Networks* (CNN) for P.o. detection with considerable success rates. An inconvenience of these approaches is that the classification is not made over the image as a whole, instead, the image is sub-divided into patches, which are later classified as P.o. or background. This approach may lead to information loss, as the classification of a patch is imposed to all its pixels.

The innovations that this work represents with respect to recent techniques in automatically identifying P.o. are: 1) the usage of a more complex deep neural network architecture that, alongside with 2) a classification by means of semantic segmentation, allows a 3) full-image pixel wise segmentation instead of a patch-based one, with no information loss or post processing needed. Finally, as a result of the aforementioned features, 4) a better *accuracy* is achieved in the classification task.

Our goal is to automatically perform a high-precision P.o. meadow segmentation in sea-floor images gathered by a bottom-looking camera mounted on an AUV, to assess its state and evolution over time. Also, we aim to execute the neural network on an AUV, passing the segmented images to an algorithm to generate real-time semantic coverage maps of P.o. areas. These maps can be used in a dynamic path planning context to adapt the vehicle trajectory, in order to optimize the mission, in terms of duration, quality and quantity of the gathered data.

This document is structured as follows. Section 2 exposes the deep network architecture used and its characteristics. Following, Section 3 describes the different study cases, containing the data acquisition, processing, model tuning and validation process. Classification results are presented in Section 4. Finally, Section 5 explains the network implementation in the AUV.

2. Deep Learning Approach

In the last few years, the new deep learning approaches have offered major improvements in *accuracy* in many computer vision tasks [27]. Causes of this are: the existence of more data, increased computation power and the development in the network architectures, making deep learning [12] one of the leading approaches in the field of computer vision.

*Corresponding author
miguel.martin@uib.es (M. Martin-Abadal); Eric Guerrero-Font (M. Martin-Abadal)

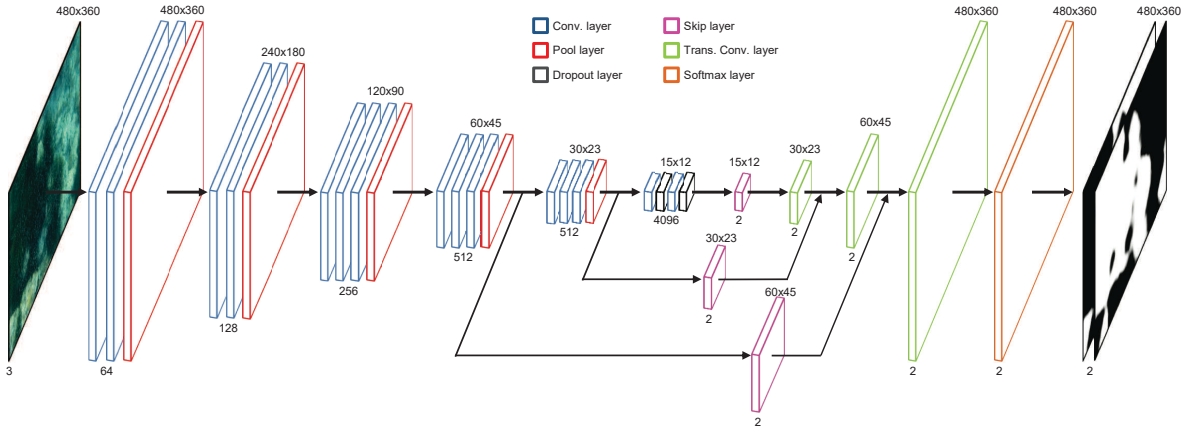


Figure 1: Neural network architecture. Encoder: convolutional (blue), pooling (red) and dropout (black) layers. Decoder: skip (purple), transposed convolutional (green) and softmax (orange) layers. The numbers under and above the layers indicate the number of feature maps and its size, respectively.

In this work we use a semantic segmentation algorithm, based on a deep neural network, in order to achieve a segmentation of the P.o. meadows. The following subsections explain the network architecture and the training details.

2.1. Network Architecture

The architecture can be divided into two main blocks, the encoder and the decoder.

2.1.1. Encoder

The encoder purpose is to extract features and spatial information from the original images. For this task, we make use of the VGG16 architecture [28], taking out the last classification layer. This architecture uses a series of convolutional layers to extract the features, along with max pool layers to reduce the feature maps dimension. Additionally, the last two fully connected layers of the VGG16 architecture are converted into convolutional layers, in order to preserve the spatial information and obtain a first low resolution segmentation.

2.1.2. Decoder

For the decoder, we use the FCN8 architecture [15]. The decoder takes the output from the last convolutional layer of the encoder and up-samples it using transposed convolutional layers [8]. Also, skip layers are utilized to combine low level features from the encoder with the higher coarse information of the transposed convolutional layers. Finally, a softmax layer is applied to obtain the prediction probability for our two classes, background and P.o. The network architecture is shown in Figure 1.

This architecture, henceforth referred as VGG16-FCN8, has already presented great results in other segmentation tasks, like class segmentation of the PASCAL VOC 2011-2 dataset in [15], or road segmentation for autonomous drive in [31].

2.2. Training Details

The VGG16-FCN8 architecture can be trained on a single forward-backward pass. The training of the encoder is performed by readjusting the kernel values in the convolutional layer filters. The decoder is trained by means of the transposed convolutional and skip layer filters.

In order to train the network we need a set of images containing P.o., and the corresponding label map of each image, where P.o. and background areas are marked in different color codes, acting as ground truth.

We use a cross-entropy loss function to train the network [4], which loss increases as the predicted probability diverges from the actual label, along with the Adam optimizer [14]. Also, dropout layers with a 0.5 probability are applied to both fully connected layers of the encoder, to prevent overfitting [29].

The encoder is initialized using pretrained VGG weights on ImageNet [6]. For the decoder, the transposed convolution layers are initialized to perform bilinear upsampling. For the skip connections we apply a truncated Gaussian initialization with low standard deviation. These configuration parameters and initialization methods have already been tested, presenting great results in [31].

3. Experimental Framework

This section exposes the whole experimental framework. First, it explains the acquisition and labelling of the images conforming the different datasets, along with its organization and usage. Next, the different study cases and hyperparameters used are presented. Finally, it describes the validation and evaluation details.

3.1. Datasets

3.1.1. Acquisition

The images are extracted from several video sequences obtained using three different cameras mounted alternately on an AUV: a GoPro, a stereo pair composed by two Manta

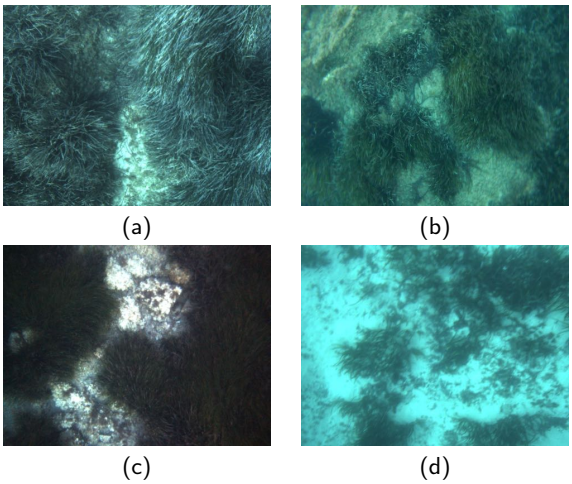


Figure 2: Images from different missions showcasing different P.o. and water conditions.

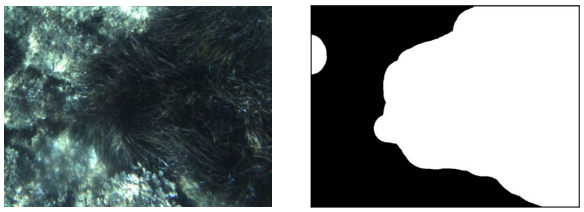


Figure 3: (a) Original image. (b) Corresponding manually generated ground truth label map, P.o. is marked in white and background in black.

G283 cameras perfectly synchronised and a Bumblebee2 rewire stereo rig, always facing downwards and with the lens axis perpendicular to the vehicle horizontal axis. The original image resolution is normalized and decimated to $480 \times 360 pixels$ for the tests presented in this work. This reduction of the image size accelerates the segmentation process considerably, permitting its execution online. The AUV specifications and the online implementation are further developed in Section 5.

Several missions were conducted on P.o. colonized coastal areas of the west and north-west of Mallorca. The objective was to obtain datasets under different P.o. conditions such as meadow density, coloration (it changes with the season and its life cycle) and health state; or water illumination, depth and turbidity, in order to build varied datasets to train and test the neural network. In all missions, the robot was programmed to move at a constant navigation altitude.

Figure 2 shows sample images from different missions showcasing different P.o. and water conditions.

3.1.2. Labeling

Label maps are built, manually, from the images gathered by the AUV. These label maps act as ground truth, in which the areas where P.o. is present are marked in white and the background areas in black. Figure 3 shows an original image along with its ground truth label map. It should be noticed that the boundary of the P.o. meadows is not well defined, making it hard to exactly determine the boundaries between the background and P.o. classes.

3.1.3. Dataset Managing

We dispose of six datasets, each one built with images extracted from video sequences recorded during different immersions, selecting sufficient images that are representative of all the aforementioned hardware and environmental conditions. We gathered one dataset from the Palma Bay, containing 164 images; another from Cala Blava, with 30 images; and four more from the Valldemossa port, of 157, 68, 41 and 23 images, respectively.

Table 1
Dataset managing.

Dataset	Location	Camera	N img.	Set
1	Palma Bay	Manta G238	164	*mix*
2	Cala Blava	Manta G238	30	*mix*
3	Valldemossa	GoPro	157	*mix*
4	Valldemossa	Manta G238	68	*mix*
5	Valldemossa	Manta G238	41	*mix*
6	Valldemossa	BumbleBee2	23	*extra*

From all these datasets, two main sets of images are generated, the mix set, including 460 and the extra set with 23 images. Table 1 indicates the location, camera used, number of images and the corresponding set of each dataset.

The mix set (460 images) is used to train and test the network, offering a wide range of diverse and different textures containing P.o. and thus, assuring robustness in the training and model selection process and also in later classification stages

The extra set (23 images) was grabbed with a camera different from the others used to grab the videos that form the mix set, it can be used as an additional test set, allowing us to detect over tting during the training and to assess how well the trained network generalizes on images acquired with a different camera and distinct unseen environmental conditions.

3.2. Study Cases

When training a neural network, there are parameters which can be tuned, changing some of the features of the network or the training process itself. These are the so called hyperparameters. In order to find the values of these hyperparameters that offer the best performance, we train the network with different values and combinations, which are shown in Table 2.

First, we train our network with and without implementing data augmentation. Data augmentation is a technique used to reduce overfitting. It consists of applying contrast and brightness changes to the training images. Therefore, the network trains over more diverse data, being able to perform better on unseen conditions. On the other hand, data augmentation may cause some *accuracy* loss on training-like

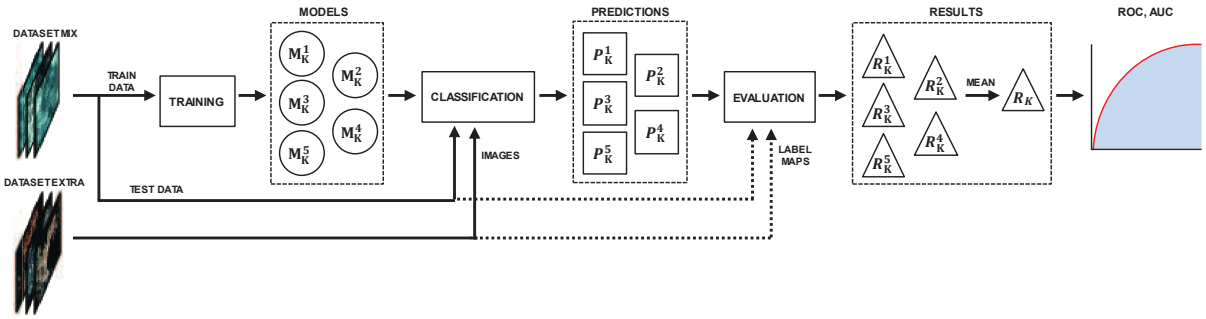


Figure 4: Experiment "K" validation process. For each one of the eight study cases, the network is trained five times using the k-fold cross-validation method, outputting five models. These models are run and evaluated over the mix and extra test sets. Finally, the ROC curve and AUC value are calculated from the five models mean performance.

Table 2
Study Cases.

Case	Data aug.	Learning rate	Iterations
1	0	1e-05	8k
2			16k
3		5e-04	8k
4			16k
5	1	1e-05	8k
6			16k
7		5e-04	8k
8			16k

images, due to the fact that the network losses specificity during the training process [30].

After, we set up two different learning rates. The learning rate value affects the size of the steps the network takes when searching for an optimal solution. Higher learning rates are able to converge quickly, but may overshoot the optimal point. In opposition, lower learning rates converge slowly, and may not be able to get to the optimal point [2].

Finally, we stipulate two different values for the number of iterations. This parameter sets the number of times the network backpropagates and trains. A higher number of iterations may get a better result over the training data, but also can overfit it, while fewer iterations may not be enough to reach the optimal point [2].

3.3. Validation

3.3.1. Validation Process

We conduct eight different experiments, each one assessing the performance of a study case.

For each experiment, the network is trained using the corresponding study case hyperparameters. To do so, we make use of the k-fold cross validation method [10]. It consists of splitting our mix set into five equally sized subsets and train the network five times, each one using a different subset as test data and the remaining four subsets as train data. This method reduces the variability of the results, as these are less dependent on the selected test and training data, obtaining a more accurate performance estimation.

From the network training, five models are generated, M_K^i, where $K=1\dots8$ represents the experiment number and $i=1\dots5$ the model index. We run the five output models with their corresponding test subset and also the whole extra set, obtaining the P.o. predictions, P_K^i, of all the models on both sets. From these predictions, each model is evaluated in order to assess its segmentation performance, R_K^i. The details of this process and the evaluation metrics are explained in Subsection 3.3.2. Finally, the segmentation performance, R_K, of each experiment is computed as the mean of its five models performance, R_K^i.

From the obtained results, we generate a *Receiver Operating Characteristic* (ROC) curve [21]. ROC curves represent the *recall* against *fall-out* values (see equations 3 and 4) of a binary classifier at various threshold settings over the probabilistic output. We also analyse the *Area Under the Curve* (AUC) of the ROC curve, which gives a quantitative measure of the classifier performance. This value ranges from 0.5 to 1.0, and grows as the ROC curve is shaped to the left (low *fall-out*) top (high *recall*) corner [13].

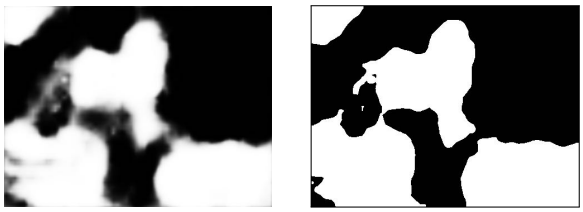
The workflow of the validation process of the experiments is shown in Figure 4.

3.3.2. Model Evaluation Details

In order to evaluate the performance of a model, we convert the probabilistic output of the softmax layer, into a binary classification image (Figure 5). The output of the model is binarized at nine equally distributed threshold values, $j=1\dots9$.

The binarized outputs of the model are compared with the corresponding ground truth label maps. For this task, we propose a simple pixel wise comparison, analysing for each pixel if the model classification output is equal or different to its corresponding ground truth label.

From this comparison, a confusion matrix is generated, indicating the number of pixel correctly identified as P.o., *True Positives* (TP), and as background, *True Negatives* (TN); and the number of pixels wrongly identified as P.o., *False Positives* (FP), and as background, *False Negatives* (FN).



(a) (b)

Figure 5: (a) Probabilistic output and (b) its corresponding binarized image.

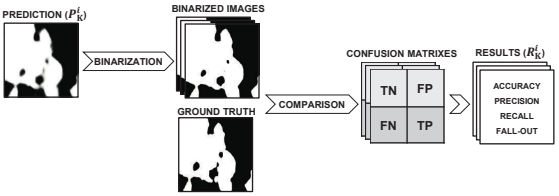


Figure 6: Model "i" of experiment "K" evaluation process. For each model, the output prediction is binarized at *j=1...9* threshold values. From every binarization *"j"*, a confusion matrix is constructed and the *accuracy*, *precision*, *recall* and *fall-out* values are calculated.

The TP, TN, FP and FN values are used to calculate the *accuracy*, *precision*, *recall* and *fall-out* of the model, defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

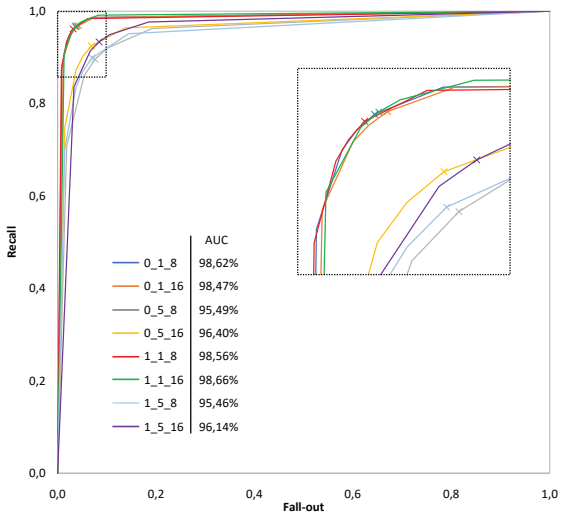
$$Fall\text{-}out = \frac{FP}{FP + TN} \quad (4)$$

Accuracy is defined as the percentage of correct pixel classifications over all classes. *Precision* represents the percentage of TP classifications with respect to all the pixels classified as positives. *Recall* refers to the percentage of TP classifications with respect to all the truly positive pixels. *Fall-out* denotes the percentage of FP classifications with respect to all the truly negative pixels.

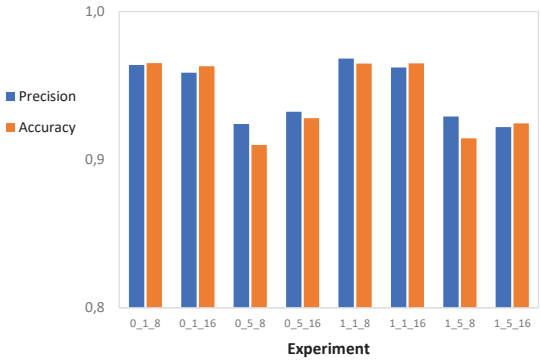
The process followed in order to determine the segmentation performance of a model is represented in Figure 6.

4. Classification Results

This section shows the results obtained for each experiment in both test sets (mix and extra), along with the hyperparameter selection process to build our final model. Finally, we perform a comparison of the selected model with other classification methods and analyse where and why the classification errors occur.



(a)



(b)

Figure 7: Results obtained from evaluating the test images of the mix set. (a) ROC curves along with their AUC values, the optimal binarization threshold for each curve is marked with an *"X"*. (b) *Precision* and *accuracy* values at the optimal binarization thresholds.

The notation used to name each experiment makes use of three numbers. The fist one refers to the data augmentation, 0 if it is not applied, and 1 if it is. The second one indicates the learning rate value, 1 if it is 1e-05, and 5 if it is 5e-04. The third one expresses the number of iterations, 8 for 8000 and 16 for 16000. For instance, the "0_1_8" experiment refers to the experiment in which data augmentation is not applied, the learning rate is 1e-05 and the network is trained for 8000 iterations.

4.1. Experiments Performance

4.1.1. mix set results

First we analyse the results obtained over the test images of the mix set. Figure 7(a) represents the ROC curve along with the corresponding AUC value of each experiment, and 7(b) shows the *precision* and *accuracy* obtained for each

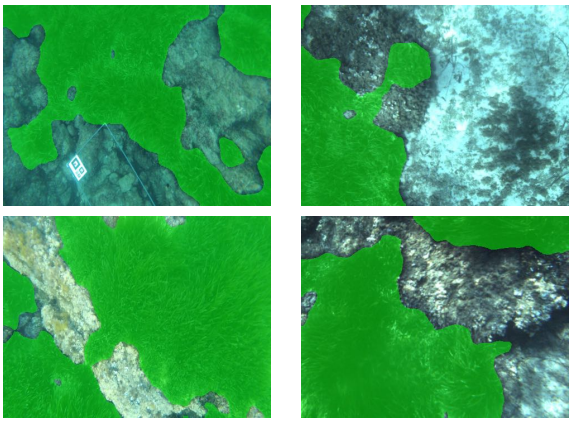


Figure 8: Visualization of the results obtained for images from the mix set. The results of the segmentation are superimposed, in green, to the original images.

experiment at its optimal binarization threshold, selected as the one with the best (higher) trade-off between recall and fall-out, calculated as:

$$Trade\text{-}off = \frac{Recall + (1 - Fall\text{-}out)}{2} \tag{5}$$

All ROC curves have an AUC over 95%, reaching a maximum of 98.7% for the 1_1_16 experiment. Following the criteria established in [20], these AUC values represent excellent classifiers.

The results show that the *precision* and *accuracy* values at optimal thresholds are greater than 90% for all the experiments. For the *precision*, the highest point is 96.5%, achieved in experiment 1_1_16, while the lowest one is 91.0%, obtained in experiment 0_5_8. For the *accuracy*, the highest point is 97.5%, achieved in experiment 1_1_8, while the lowest one is 92.2%, obtained in experiment 1_5_16.

Experiments with the higher learning rate present slightly worse *precision*, *accuracy* and AUC values than the experiments with the lower one. On the contrary, neither the number of iterations nor the application or not of data augmentation have a significant impact on the performance.

Qualitative results of the segmentation over the mix set are shown in Figure 8.

4.1.2. extra set results

While the results over the test data of the mix set are promising, as mentioned in Subsection 3.1.3, the test images are from the same immersions as the images used for the training and thus, the environmental conditions are similar. In order to assess the performance of the classifiers on unseen conditions, we analyse the results over the extra set, which are shown in Figure 9.

The AUC values are significantly lower for the experiments with the higher learning rate, around 92%, independently of the data augmentation state or the number of iterations. Otherwise, the experiments with the lower learning rate are able to maintain similar results as the previous

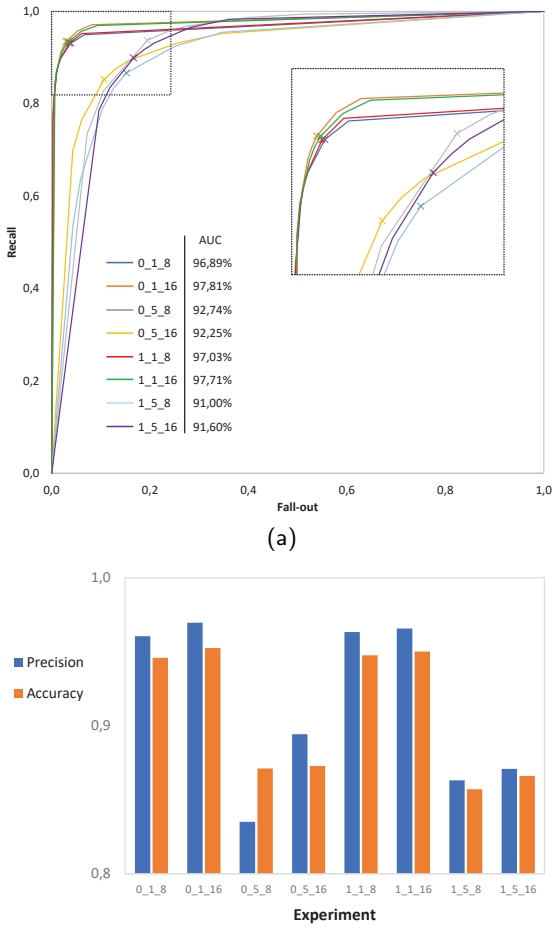


Figure 9: Results obtained from evaluating the test images of the extra set. (a) ROC curves along with their AUC values, the optimal binarization threshold for each curve is marked with an *"X"*. (b) *Precision* and *accuracy* values at the optimal binarization thresholds.

test, reaching values around 97.7% when performing 16000 iterations and 97.0% when 8000. This means that these experiments do not overfit the training data, generalizing their training well enough to still perform a good classification even on images obtained with a different camera and environmental conditions that have not been trained on.

This can also be noticed by looking at the *precision* and *accuracy* values, calculated at the optimal binarization threshold for each experiment. The experiments with the higher learning rate achieve values around 85% for both metrics. For the experiments with the lower learning rate, the *precision* and *accuracy* values are around 96% and 95%, respectively. Again, the experiments performed with 16000 iterations have a slightly higher *precision* and *accuracy* values, while the effect of applying data augmentation or not is negligible.

Qualitative results of the segmentation over the extra set are shown in Figure 10.

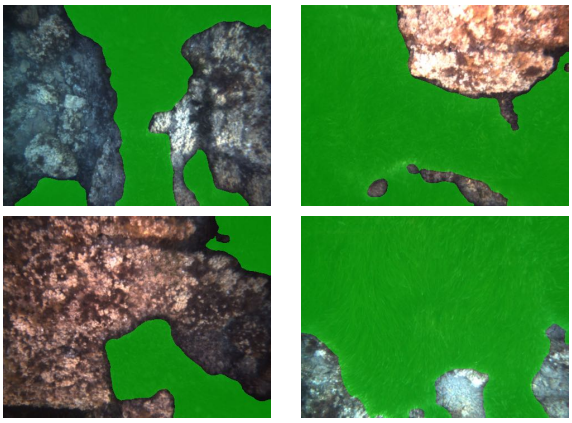


Figure 10: Visualization of the results obtained for images from the extra set. The results of the segmentation are superimposed, in green, to the original images.

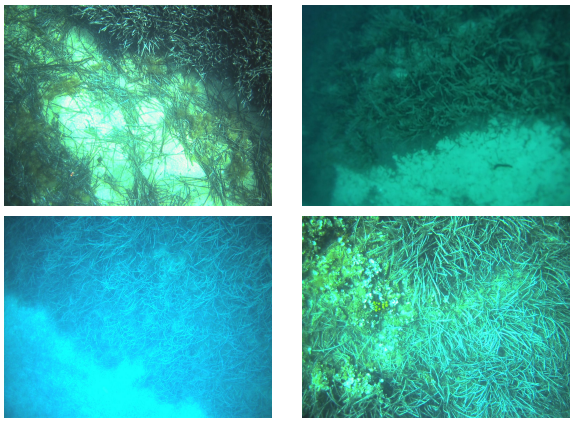


Figure 11: Images from the croatia test set.

4.2. Hyperparameter and Model Selection

4.2.1. Hyperparameters selection

As a result of evaluating all experiments on both test sets, we can select the hyperparameters that show better performance.

Firstly, we select a learning rate of 1e-05. The results obtained on both mix and extra tests clearly show that the experiments with the lower learning rate obtain better AUC, *precision* and *accuracy* values.

Secondly, we decide to train with 16000 iterations. In the mix results we can observe that, among the lower learning rate experiments, those with a larger number of iterations have a slightly better performance.

Finally, we opt to apply data augmentation in order to generalize the training to future immersions with new unseen environmental conditions. The results show that applying it does not incur in a worse classification over the test data.

4.2.2. Model selection

We make an in-depth study of the performance variability for the aforementioned selected hyperparameters by re-conducting ten times the validation process exposed in Subsection 3.3, obtaining a total of fifty output models. After evaluating all models, we carry out an statistical analysis, computing the *mean* and *standard deviation* (*std*) of the *precision* and *accuracy* over both test sets altogether.

For the *precision*, the mean is 96.95% with a *std* of 0.97%. For the *accuracy*, the mean is 96.08% with a *std* of 0.49%. Such low *std's* indicate that all fifty models show a very similar performance around the mean, meaning that our network architecture and validation process are robust.

Afterwards, the model with best performance is selected from the previous fifty. This final model has a *precision* of 96.57% and an *accuracy* of 96.81%. This is the selected model to perform the online segmentation in the AUV.

4.3. Comaprison

In this section we present a comparison of the VGG16-FCN8 architecture with the classification methods mentioned in Section I, the Burguera et al. method [3] (henceforth ML-SVM) and the Gonzalez-Cid et al. method [11] (henceforth CNN), as well as to other state-of-the-art semantic segmentation architectures such as the U-Net [24] and the SegNet [1]. The performance comparison is conducted using the evaluation metrics de ned in Section III-C.2, which are obtained from the classification of the images pertaining to three test sets.

The first test set is the already known extra set, which contains images with new and unseen water and P.o. conditions for the classifiers.

The second test set (henceforth, croatian set) was provided by the "Laboratory for Underwater Systems and Technologies" research group, at the University of Zagreb. It consists of 23 images extracted from video sequences recorded using a lightweight AUV by Ocenascan-MST and a Lumenera Le165 camera during different immersions in the Peljesac peninsula, Croatia. Figure 11 shows images from this test set.

Finally, the third test set (henceforth, islands set) was provided by the "Ecología Interdisciplinaria" research group, at the University of the Balearic Islands. It consists of 27 images extracted from video sequences recorded by scuba-divers using a GoPro camera during different immersions in the Mediterranean islands of Ibiza, Formentera and Menorca. Figure 12 shows images from this test set.

The croatian and islands test sets represent a challenge for the classifiers, as they were taken in new locations, following different recording procedures and using different cameras, thus, the images of these new test sets contain distinct water and P.o. conditions. Besides, the images were taken at a different distance to the P.o. meadows and with a different angle respect the sea-floor, facts that also may condition the classifiers performance.

These three sets allow us to further test the robustness of the classifiers and check their capability to be used in external applications.

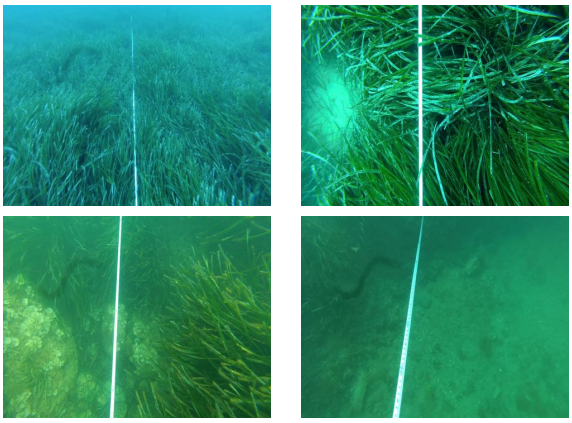


Figure 12: Images from the islands test set.

Table 3
Classification performance comparison over the extra test set.

Method	Acc.	Prec.	Recall	Fall-Out
ML-SVM	89.1%	87.1%	94.9%	*18.0%*
CNN	62.2%	81.0%	31.9%	*7.5%*
U-Net	93.1%	93.9%	92.1%	*6.0%*
SegNet	90.0%	90.4%	91.5%	*9.7%*
VGG16-FCN8	**96.1%**	**97.2%**	**95.0%**	***2.8%***

Table 4
Classification performance comparison over the croatian test set.

Method	Acc.	Prec.	Recall	Fall-Out
ML-SVM	66.9%	75.0%	37.9%	*10.0%*
CNN	62.0%	79.7%	32.1%	*8.2%*
U-Net	82.3%	83.2%	81.0%	*16.4%*
SegNet	83.2%	73.5%	82.7%	*16.3%*
VGG16-FCN8	**94.0%**	**93.7%**	**94.4%**	***6.4%***

For the ML-SVM method, we use the model trained over color images downsampled to 160x120 pixels and using 32x24 pixels patches, which was one of the parameter combinations that showed best results.

For the CNN method, we select the model trained using a learning rate of 1e-03 for 10 epochs with a batch size of 100.

Finally, for all semantic segmentation methods (VGG16-FCN8, U-Net and SegNet) we train them using the selected hyperparameters in Section IV-B.1 and the data from the mix set.

Tables 3, 4 and 5 show the results of the evaluation metrics of all compared classification methods over the extra, croatian and islands test sets, respectively.

We can notice that the CNN method is the worst one in all test sets, mainly due to the patch-wise classification. The ML-SVM method seems to have been designed to be conservative when classifying the P.o. As a result, when it classifies a pixel as P.o., it is highly likely it is P.o., but the Recall and Fall-Out values denote that several pixels that truly are P.o. will be classified as background.

Table 5
Classification performance comparison over the islands test set.

Method	Acc.	Prec.	Recall	Fall-Out
ML-SVM	65.7%	88.6%	59.5%	*19.0%*
CNN	67.6%	65.7%	73.9%	*38.6%*
U-Net	81.2%	81.2%	81.0%	*18.7%*
SegNet	70.3%	70.4%	69.8%	*29.3%*
VGG16-FCN8	**87.6%**	**86.4%**	**89.2%**	***14.0%***

Consequently, it can be noticed that the ML-SVM method has a slightly better Precision than the VGG16-FCN8 when classifying the croatian and islands test sets, but the Recall and Fall-Out values are significantly worse. On the contrary, VGG16-FCN8 presents good values in the four metrics, which implies that it is a better classifier for both P.o. and background pixels.

On the other hand, considering the three semantic segmentation classifiers, the U-Net and SegNet methods have a similar performance when classifying extra and croatian test sets, while U-Net shows better results when classifying the island test set. VGG16-FCN8 presents the best results of the three, suggesting again being the best semantic segmentation classifier.

To sum up, after comparing 5 different classifiers over 3 different sets of P.o. underwater images, the classifier that presents better results in terms of the four evaluation metrics: Precision, Accuracy, Recall and Fall-Out, is the one presented in this paper VGG16-FCN8, indicating that it is the most robust and the best option for P.o. classification in underwater images.

4.4. Error Analysis

To train and evaluate the VGG16-FCN8 network we have made use of labelled images, manually generating the ground truths. This is a tedious task, subject to errors. Being aware that the evaluation of the results of the VGG16-FCN8 method could depend on the small errors present in the ground truth images, this section aims to analyse where and why the classification errors occur.

In order to do carry out this analysis we evaluate the mix set test images with the selected nal model from Section 4.2.2. The error analysis is conducted from the binarization of the probabilistic output at the optimal threshold.

Firstly, we perform a comparison between the binarized output and the corresponding ground truth images. The areas where these two images do not match are the FP and FN classifications. Figure 13 shows a superposition of an original image with the aforementioned comparison, marking the FN in blue and the FP classifications in green.

The majority of the errors are located on the boundaries of the P.o. meadows. As stated in Subsection 3.1.2, the boundary of the P.o. meadows is not well defined and hard to determine exactly, even during the manually ground truth generation process.

In order to determine if these FN and FP are really classification errors or a ground truth labeling issue, we

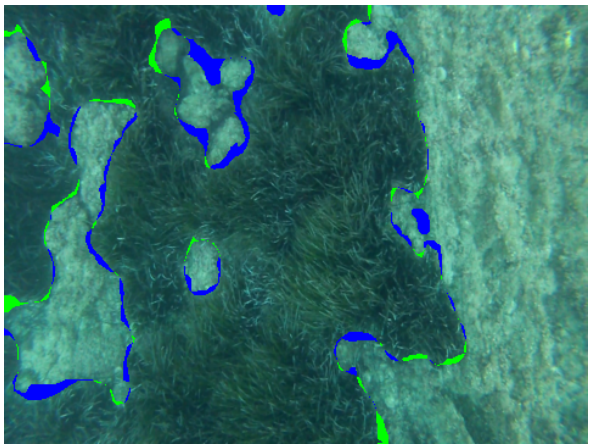
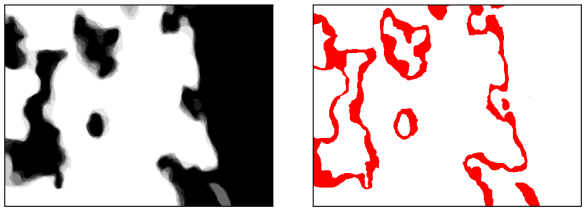


Figure 13: Superposition of an original test image with the computed error, generated by comparing the network output with the image ground truth label map. FN are marked as blue and FP as green.



(a) (b)

Figure 14: (a) Mean of the manually marked label map and. (b) Area of uncertainty of the hand labeled ground truth, obtained as the area where not all ground truths match.

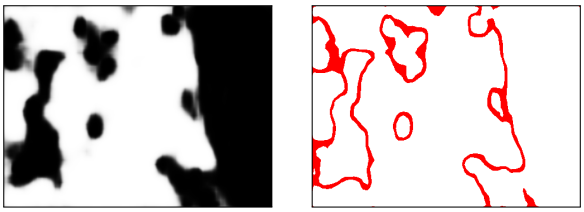
decide to calculate the area of uncertainty of the hand labeled ground truth and see if the errors are included in it.

To do so, we ask ten people to generate the label maps of the testing images (without including the one who has generated the ground truth used to assess the network classification). Then, we compute the mean grey level for each pixel of these label maps. The areas where not all ground truth match, are marked as areas of uncertainty.

Figure 14(a) shows the computed mean label map, and 14(b) shows the obtained area of uncertainty for the original image shown in Figure 13 .

For this image, a 94.6% of the misclassified pixels fall into the area of uncertainty of the hand labeled ground truth. From this, we can infer that most of the network errors do not come from misclassified pixels, but from the ground truth labeling process.

Finally, we also calculate the area of uncertainty of the neural network output as the difference in classification between using 1% and 99% threshold values. This means that the uncertainty area is conformed by the pixels that the network is not entirely sure if they belong to the P.o. or background class.



(a) (b)

Figure 15: (a) Probabilistic output of the network. (b) Area of uncertainty of the neural network, obtained as the classification difference when using a very high and a very low threshold.

Figure 15(a) shows the probabilistic output of the net when evaluating the case study image, and 15(b) shows its corresponding area of uncertainty of the neural network.

For this image, the area of uncertainty presented by the network represents an 18.9% of the whole image, while the one from the hand labeled ground truth is bigger, representing a 28.5%. As can be seen, both areas of uncertainty present a very similar shape, located on the boundaries of the P.o. meadows.

These factors, along with the fact that most FN and FP are included in the uncertainty area, means that the network output is more reliable than the manually generated ground truth label map.

5. AUV Implementation

The objective of this section is to describe the implementation of the semantic segmentation network in the AUV and its online execution, using it to generate real-time semantic coverage maps of P.o. meadows. This is carried out by surveying the area of interest with an AUV and recording images and their geolocalization, then, these images are processed and segmented online and passed to the coverage map generation algorithm.

In this section we present an overview of the used AUV characteristics and navigation, and the implementation of the neural network in the AUV used to perform online segmentation during the robot operation.

5.1. Turbot AUV

The Turbot AUV (Figure 16), property of the University of the Balearic Islands, is a SPARUS II model unit [5]. It is equipped with three motors which grant it three degrees of mobility (surge, heave and yaw). Also, it has a navigation payload, composed by: 1) a DVL (Doppler Velocity Log) to get linear and angular speeds and altitude, 2) a pressure sensor to get high frequency depth measurements, 3) an IMU (Inertial Measurement Unit) to measure accelerations and angular speeds, 4) a Compass for heading, 5) a GPS to be geo-referenced during surface navigation, and 6) an USBL (Ultra Short Baseline) acoustic link used for localization and data exchange between the robot and a remote station.

Furthermore, a stereo pair of *Point Grey CM3-U3-31S4* cameras facing downwards provides the robot with images of $2048 \times 1536 pixels$ resolution. These images are mainly

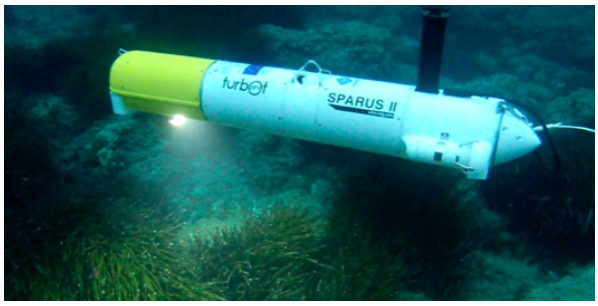


Figure 16: Turbot AUV: SPARUS II.

used for three purposes: *a*) getting visual odometry (altitude and linear and angular speeds), *b*) performing online P.o. segmentation, and *c*) mapping the surveyed area.

The robot has two computers. One is dedicated to capturing and processing the navigation sensor data and running the main robot architecture, which is developed under the ROS middleware [22]. The second computer is where the image grabbing and online segmentation processes are executed, its speci cations are: Intel i7 processor working at 2.5 GHz, 4 cores, 8GB of RAM and Ubuntu 16.04 O.S.

To perform a survey mission the vehicle must have a good estimation of its localization -*Where am I?*-, a well defined mission -*Where should I go?*-, and a proper path planning approach -*How do I get there?*-.

The localization of the vehicle is obtained through the fusion of multiple state estimations produced by the DVL, IMU, Compass, GPS, USBL, visual odometry and a navigation filter [9]. The survey mission is defined with a series of waypoints programmed to cover all the desired region, and with a given altitude, usualy ranging between 2 and 4m, conditioned by the water turbidity, lighting conditions and the vehicle cruise speed. Finally, for the sake of simplicity, the strategy used by the AUV to get to the planned waypoints is a *Line Of Sight* (LOS) method applied to control the horizontal position using two lateral thrusters, and an altitude control using the vertical motor located at its gravity center.

5.2. Online Image Segmentation

5.2.1. Implementation

To perform the online segmentation we implement a pipeline based on ROS. It loads a frozen inference graph of a trained model and executes two threads; one for the image gathering and another for the image segmentation.

The image gathering thread codifies every input image to *RGB* and then rectifies and decimates them to $480 \times 360 pixels$. The image segmentation thread receives the images and feeds them into the frozen inference graph, which generates the online P.o. segmentation.

5.2.2. Experiments

The experiments were conducted on the north coast of Mallorca, in shallow waters of 6m depth. The AUV operated at a velocity $v = 0.4m/s$ and a navigation altitude $a = 2.5m$.

In order to perform the segmentation of the images, it was used the frozen inference graph of the model that has shown the best performance (selected in Subsection 4.2). The obtained segmentation framerate was 0.42 FPS.

An illustrative video showing the online segmentation can be seen on the SRV group web page [18]. The video shows, at the left of the screen, the video sequence captured from the camera, and at the right, the results of the segmentation superimposed in green to the original frames.

5.2.3. Validation

The performance is analysed in terms of the obtained framerate of the output segmentation stream. The only requirement is that, in order to avoid gaps in the generation of semantic coverage maps, the successive segmented images need to overlap.

This overlap depends on the camera displacement between two consecutive *keyframes* (d_{KF}) and on the height of the image footprint (h_{FP}). Then, the *overlap* can be expressed as:

$$overlap = (h_{FP} - d_{KF}) \cdot h_{FP}^{-1} \tag{6}$$

$$d_{KF} = v \cdot framerate^{-1} \tag{7}$$

$$h_{FP} = (a \cdot h_{image}) \cdot f^{-1} \tag{8}$$

Where v denotes the AUV velocity, a the navigation altitude, h_{image} the image height pixels and f the focal length.

Using the aforementioned vehicle speed and navigation altitude, along with an image height resolution of $h_{image} = 360 pixels$, a focal length of $f = 623.3p$, and the obtained segmentation framerate. The resulting overlap is 34.0%. Thus, the framerate is high enough to get images overlap.

6. Conclusion

This section enumerates the main conclusions of this work. We have used a semantic segmentation deep network architecture to automatically perform P.o. classification on underwater images. The obtained results showed (1) very high levels of *accuracy* for diverse hyperparameter configurations, the highest one was achieved when data augmentation was applied and the network was trained with a learning rate of 1e-05 for 16000 iterations. Also, the low *std* of the evaluation metrics indicates that (2) our architecture and evaluation process are robust.

The error analysis showed that most misclassified pixels fall into the uncertainty area of the manually generated ground truth label maps. This is due to the ground truth issues caused by the fuzzy boundaries of P.o., inferring that the classification performance might be even better than the one shown on the results of the validation process.

This, along with the fact that the uncertainty area of the network is smaller than the one from the hand labeled ground truth, means that (3) the reliability of the network was higher that the manually labeling process.

Finally (4), we have implemented the segmentation process running online in an AUV operating in real environments. From the validation we obtained that the framerate of the segmented images was high enough to get images overlap, permitting an adequate semantic mapping of PO meadows.

Further developments will focus on lightening the online segmentation computational load while maintaining high *accuracy* levels. The aim is to provide more computational power to forthcoming autonomous exploration techniques like online mission replanning. Also, we will consider a multi-class classification, differentiating between diverse algae types and backgrounds such as rocks or sand.

The code containing the network architecture and its training process, along with the used datasets and the codes to perform the images preprocess, output validation and error analysis, are available on a GitHub repository [17].

Acknowledgments

This work is partially supported by Ministry of Economy and Competitiveness, under contracts TIN2014-58662-R, DPI2014-57746-C3-2-R and TIN2017-85572-P, for all of them (AEI/MINECO/FEDER, UE); by SOIB, under the JQ-SP 49/17 project (ESF, Youth Guarantee); and by the Government of the Balearic Islands through grant FPI/2031/2017 (Vicepresidencia i Conselleria d'Innovació, Recerca i Turisme).

CRediT authorship contribution statement

Miguel Martin-Abadal: Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. **Miguel Martin-Abadal:** Methodology, Software, Investigation, Data Curation, Writing - Review & Editing. **Francisco Bonin-Font:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Yolanda Gonzalez-Cid:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

[1] Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 2481–2495. doi:doi: 10.1109/TPAMI.2016.2644615.

[2] Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. Neural Networks: Tricks of the Trade .

[3] Bonin-Font, F., Burguera, A., Lisani, J.L., 2017. Visual discrimination and large area mapping of posidonia oceanica using a lightweight auv. IEEE Access 5, 24479–24494.

[4] Buja, A., Stuetzle, W., Shen, Y., 2005. Loss Functions for Binary Class Probability Estimation and Classification: Structure and Applications. Technical Report. Deptartment of Statistics, University of Pennsylvania.

[5] Carreras, M., Hernández, J.D., Vidal, E., Palomeras, N., Ribas, D., Ridao, P., 2018. Sparus II AUV - A hovering vehicle for seabed inspection. IEEE Journal of Oceanic Engineering 43, 344–355. doi:doi: 10.1109/JOE.2018.2792278.

[6] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F., 2009. Imagenet: a large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition , 248–255.

[7] Diaz-Almela, E., Duarte, C., 2008. Management of Natura 2000 Habitats 1120, (Posidonia Oceanicae). Technical Report. European Commission.

[8] Dumoulin, V., Visin, F., 2016. A guide to convolution arithmetic for deep learning. ArXiv e-prints - 1603.07285 arXiv:1603.07285.

[9] Font, E.G., Bonin-Font, F., Negre, P.L., Massot, M., Oliver, G., 2017. USBL Integration and Assessment in a Multisensor Navigation Approach for field AUVs. International Federation of Automatic ControlIFAC (IFAC) 50, 7905–7910.

[10] Geisser, S., 1975. The predictive sample reuse method with applications. Journal of the American Statistical Association 70, 320–328.

[11] Gonzalez-Cid, Y., Burguera, A., Bonin-Font, F., Matamoros, A., 2017. Machine learning and deep learning strategies to identify posidonia meadows in underwater images. OCEANS 2017 - Aberdeen , 1–5.

[12] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. Http://www.deeplearningbook.org.

[13] Hanley, J., Mcneil, B., 1982. The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. Radiology 143, 29–36.

[14] Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. ArXiv e-prints - 1412.6980 arXiv:1412.6980.

[15] Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 3431–3440doi:doi: 10.1109/CVPR.2015.7298965.

[16] Marba, N., Duarte, C., 2010. Mediterranean warming triggers seagrass (posidonia oceanica) shoot mortality. Global Change Biology 16, 2366–2375.

[17] Martin-Abadal, M., 2018a. Posidonia semantic segmentation. GitHub repository URL: https://github.com/srv/Posidonia-semantic-segmentation.

[18] Martin-Abadal, M., 2018b. Video: Online Posidonia oceanica segmentation. URL: http://srv.uib.es/po-identification/.

[19] Montefalcone, M., Rovere, A., Parravicini, V., Albertelli, G., Morri, C., Bianchi, C.N., 2013. Evaluating change in seagrass meadows: A time-framed comparison of side scan sonar maps. Aquatic Botany 104, 204–212.

[20] Powers, D.M.W., 2011. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. International Journal of Machine Learning Technology 2, 37–63.

[21] Provost, F., Fawcett, T., Kohavi, R., 2001. The case against accuracy estimation for comparing induction algorithms. Proceedings of the Fifteenth International Conference on Machine Learning .

[22] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., 2009. ROS: an Open Source Robot Operating System. ICRA Workshop on Open Source Software .

[23] Rende, F.S., Irving, A.D., Lagudi, A., Bruno, F., Scalise, S., Cappa, P., Montefalcone, M., Bacci, T., Penna, M., Trabucco, B., Di Mento, R., Cicero, A.M., 2015. Pilot application of 3D underwater imaging techniques for mapping Posidonia oceanica (L.) delile meadows. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences , 177–181doi:doi: 10.5194/isprsarchives-XL-5-W5-177-2015.

[24] Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. arXiv:1505.04597.

[25] Lopez y Royo, C., Pergent, G., Pergent-Martini, C., Casazza, G., 2010. Seagrass (posidonia oceanica) monitoring in western mediterranean: Implications for management and conservation. Environmental monitoring and assessment 171, 365–80.

[26] Sagawa, T., Komatsu, T., 2015. Simulation of seagrass bed mapping by satellite images based on the radiative transfer model. Ocean Science Journal 50, 335–342. doi:doi: 10.1007/s12601-015-0031-3.

[27] Shoham, Y., Perrault, R., Brynjolfsson, E., Clark, J., LeGassick, C., 2008. Artificial intelligence index 2017 Annual Report. Technical Report. aiindex.

[28] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. ArXiv e-prints - 1409.1556 arXiv:1409.1556.

[29] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958.

[30] Taylor, L., Nitschke, G., 2017. Improving Deep Learning using Generic Data Augmentation. ArXiv e-prints - 1708.06020 arXiv:1708.06020.

[31] Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., Urtasun, R., 2016. MultiNet: Real-time joint semantic reasoning for autonomous driving. ArXiv e-prints - 1612.07695 arXiv:1612.07695.

[32] Telesca, L., Belluscio, A., Criscoli, A., Ardizzone, G., Apostolaki, E.T., Fraschetti, S., Gristina, M., Knittweis, L., Martin, C.S., Pergent, G., Alagna, A., Badalamenti, F., Garofalo, G., Gerakaris, V., Pace, M.L., Pergent-Martini, C., Salomidi, M., 2015. Seagrass meadows (posidonia oceanica) distribution and trajectories of change. Scientific reports .

[33] Vasilijevic, A., Miskovic, N., Vukic, Z., Mandic, F., 2014. Monitoring of Seagrass by lightweight AUV: A Posidonia oceanica case study surrounding Murter island of Croatia.

A deep learning solution for *Posidonia oceanica* seafloor habitat multiclass recognition

Miguel Martin-Abadal*, Ivan Riutort-Ozcariz, Gabriel Oliver-Codina and Yolanda Gonzalez-Cid

Department of Mathematics and Computer Science. University of the Balearic Islands, 07122, Palma, Spain

ARTICLE INFO

The work presented in this preprint has been published in the journal *Sensors* as:

Martin-Abadal, M.; Oliver-Codina, G.; Gonzalez-Cid, Y. *A deep learning solution for Posidonia oceanica seafloor habitat multiclass recognition*. OCEANS 2019 - Marseille, 1-7.
DOI: 10.1109/OCEANSE.2019.8867304

ABSTRACT

Recent studies have shown evidence of a significant decline of the *Posidonia oceanica* meadows on a global scale. The monitoring and mapping of these meadows and its marine habitat are fundamental tools for measuring its status and growth opportunities. The presence of hard substrates benefits *P. oceanica* survival and development rates. We present an approach based on a deep neural network to automatically perform a high-precision semantic segmentation of *P. oceanica* meadows and its seafloor habitat in sea-floor images, offering several improvements over the state of the art techniques. The presented network is able to accurately distinguish the most relevant classes: *P. oceanica* meadows, and rocky and sandy areas.

1. Introduction

Posidonia oceanica is an endemic plant of the Mediterranean sea, declared World Heritage by UNESCO in recognition to its multiple benefits to the marine and coastal ecosystems [5]. In the last decades its meadows have been degrading quickly [11][19] due to ambient conditions like global warming, or the human activity like boat anchoring [9][13]. For this reasons, the European Commission's directive 92/43/CEE defines the *P. oceanica* as a priority natural habitat.

A very important part of *P. oceanica* control and recovery comes through monitoring and mapping of its meadows. These are fundamental tools for measuring its status, helping to detect decline trends early on or address the effectiveness of any protective or recovery initiative. Furthermore, it is important to study the seafloor habitat where this plant develops, as it will provide information about its expansion opportunities and survival rate . It has been proved that due to its root system adhesive properties, the *P. oceanica* prefers hard substrates like rocky ones over the sandy ones [8][1].

Currently, these monitoring tasks are mostly carried out by divers, measuring in a manual manner meadow parameters such as lower limit depth, shoot density or extension [16]. However, the collection of these data is slow and costly [3][4]. Diverse studies have tackled the automation of this process, Recently [2] has achieved a fully autonomous detection by means of combining traditional image descriptors alongside *Machine Learning* (ML) with *Support Vector Machines* (SVM). Also, in [7] the idea of using *Convolutional Neural Networks* (CNN) for *P. oceanica* detection was explored with considerable success rates. Finally, in [12] a pixel-wise semantic segmentation of *P. oceanica* is carried out using deep learning techniques.

Nevertheless, these approaches present a variety of inconveniences. In [2] and [7] the classification is not made over the image as a whole, instead, the image is sub-divided into patches, which are later classified as *P. oceanica* or background, leading to information loss on the meadow boundaries. Moreover, in all of the previously mentioned approaches only the *P. oceanica* class was classified, without taking into account the rest of the surrounded seafloor habitat elements.

This paper presents an approach based on a deep semantic neural network to automatically perform a multiclass high-precision pixel-wise classification of sea-floor images. We established different sets of classes to train and test the network, comparing the results between them and identifying the ones that were more useful to define the *P. oceanica* meadows and its seafloor habitat.

The remainder of this paper is structured as follows. Section 2 presents the deep neural network architecture and its characteristics. Section 3 describes the adopted methodology and materials used in this work. The experimental results and discussions are presented in Section 4. Finally, Section 5 exposes the main conclusions and outlines possible future lines of work.

2. Deep Learning approach

In this work we use a semantic segmentation algorithm, based on a deep neural network, in order to achieve the classification of *P. oceanica* and its seafloor habitat.

The architecture of the network, shown in Figure 1, can be divided into two main blocks, the encoder and the decoder. The encoder purpose is to extract features and spatial information from the original images. For this task, we make use of the VGG16 CNN architecture [17], based on convolutional and pooling layers.

For the decoder, we use the FCN8 architecture [10]. The decoder takes the output from the last convolutional layer of the encoder and up-samples it using skip and transposed convolutional layers [6]. Finally, a softmax activation layer generates a gray scale confidence map for each class. These

*Corresponding author
miguel.martin@uib.es (M. Martin-Abadal)

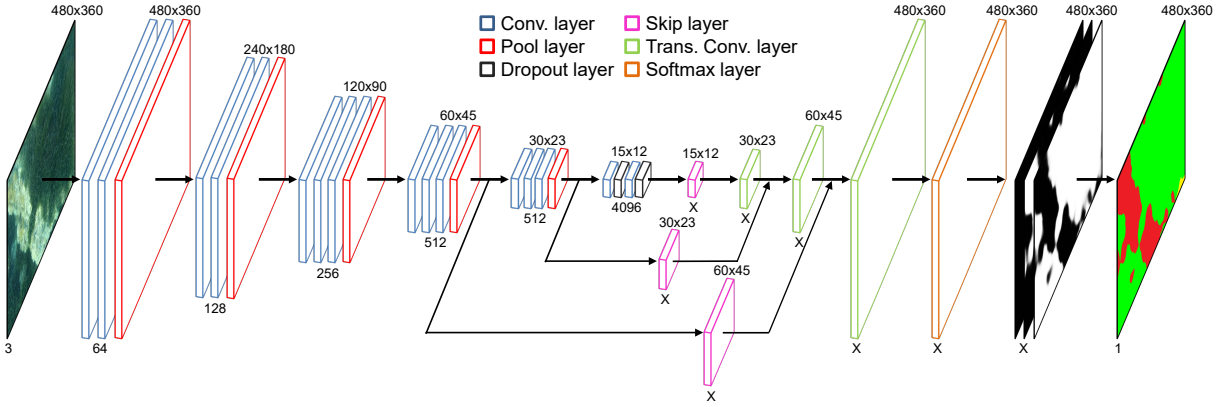


Figure 1: Neural network architecture. Encoder: convolutional (blue), pooling (red) and dropout (black) layers. Decoder: score (purple), transposed convolutional (green) and softmax (orange) layers. The numbers under and above the layers indicate the number of feature maps and its size, respectively.

maps are later converted into the final prediction by assigning to each pixel of the original image the class with highest confidence level.

The training process makes use of images containing *P. oceanica* and its seafloor habitat, along with their corresponding label maps, where each class is marked in a different colour, acting as ground truth.

For the network hyperparameters, we used a learning rate of 1e-05 and perform 16000 iterations. We also applied data augmentation. This architecture and training hyperparameters have already presented great results in other segmentation tasks, like class segmentation of the PASCAL VOC 2011-2 dataset in [10], road segmentation for autonomous drive in [18], or *P. oceanica* meadow segmentation in [12].

3. Methodology

This section explains the acquisition method and labelling process of the training and testing data. Next, the different sets of studied classes are presented. Finally, it describes the evaluation process and metrics.

3.1. Data

The required images to train and test the neural network were extracted from several underwater videos. These videos were recorded by a bottom looking camera mounted on an Autonomous Underwater Vehichle (AUV). The videos correspond to six immersions conducted on different coastal areas of Majorca island. The objective was to build a robust dataset of images gathered under different conditions such as *P. oceanica* meadow density, coloration and health state; or water illumination, depth and turbidity.

Using this method we obtained 302 images, an 80% (242 images) were used for the network training, and a 20% (60 images) to test the obtained model. From each image we generated one label map for each studied class, acting as ground truth. The areas where the corresponding class was present were marked in white, and the rest in black. From these label maps (one for each studied class that is present) we built a merged final label map for each original image, marking the areas were each of the studied classes were present with a different colour. Figure 2 shows an original image along with its ground truth label maps in black and white and the merged final label map.

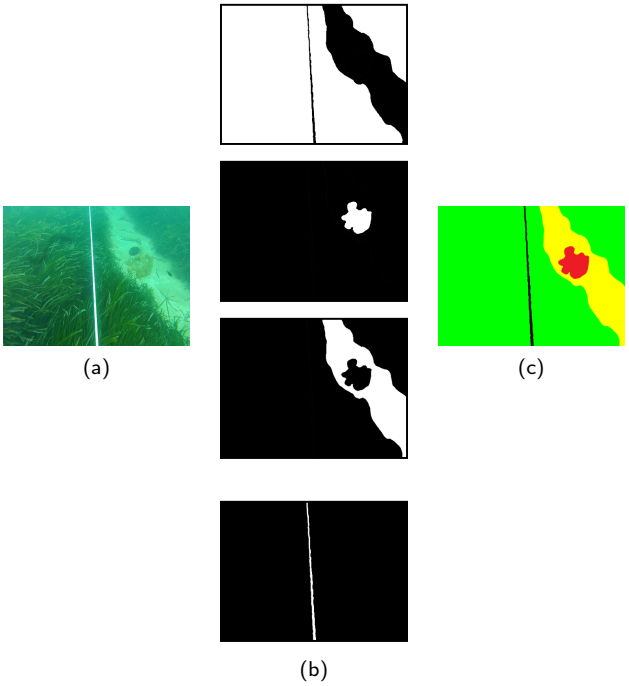


Figure 2: (a) Original image. (b) Class BW ground truth label maps. (c) Merged final RGB label map. For this set of classes: *P. oceanica* is marked in green, rock in red, sand in yellow, and background in black.

3.2. Class sets definition

The seafloor habitat in which *P. oceanica* develops are mainly sandy and rocky seafloors. In order to best describe this particular seafloor habitat we decided to distinguish

Table 1
Class sets and corresponding colour code.

	Po-a	Po-d	Rock	Sand	Matte	Back
Set 1						
Set 2						
Set 3						

between six classes, used to define the areas where different elements were present. These classes were:

- **Alive *P. oceanica* (Po-a):** healthy, grown *P. oceanica* areas.
- **Dead *P. oceanica* (Po-d):** areas with congregated dead shoots of *P. oceanica* .
- **Rock (Rock):** rocky areas.
- **Sand (Sand):** sandy areas.
- ***P. oceanica* dead matte (Matte):** areas where the *P. oceanica* has died and only its dead matte is left.
- **Background (Back):** areas containing elements different from the specified above or unrecognisable areas.

Since some of these classes look alike and present similar features, we decided to group some of them and establish three different sets of classes. On the first set (Set 1), we just contemplated four classes: Po (containing both Po-a and Po-d), Sand, Rock and Back. In this set we did not distinguished neither between alive and dead *P. oceanica* nor the areas where dead matte was present (class Matte), which were classified either as Rock or Sand. The second set (Set 2) had the same Rock, Sad and Back classes, but we distinguished between alive and dead *P. oceanica*, obtaining the Po-a and Po-d classes, respectively. Finally, the third set (Set 3) contained the Set 2 classes and also contemplated the *P. oceanica* dead matte class (Matte). The set classes along with the colour code used to generate the ground truth label maps for each set is represented in Table 1.

3.3. Evaluation

As stated in section 2, the output of the network is a grey scale map of each studied class indicating the probability of the pixel to correspond to that class. Later these predictions are merged into a final prediction, which classifies each pixel of the original image into one of the selected classes. The conducted evaluation can be splitted into two steps, the first one take into account the first predictions, and the second one the merged final prediction.

The first step focuses in the comparison between the raw grey scale maps outputted by the network for each class and their corresponding BW ground truth label maps. We generated a Receiver Operating Characteristic (ROC) curve [15] for each class. The ROC curve represents the recall against fall-out values of a binary classifier when performing a gray level threshold sweep over the probabilistic output.

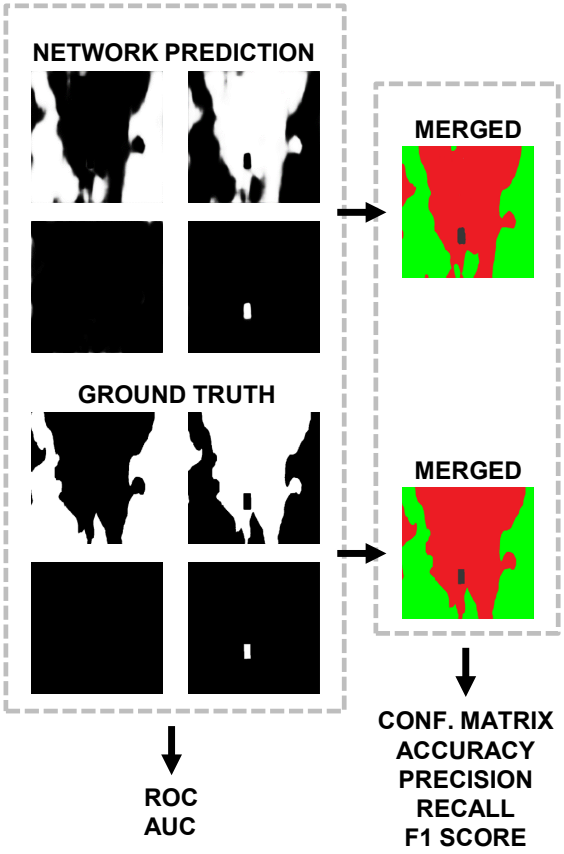


Figure 3: Evaluation workflow.

Finally, we calculated the Area Under the Curve (AUC) of the ROC curve. This metric gives information on how well the network was able to identify each class separately, without interfering and overlapping one with each other.

The second evaluation step is targeted on the comparison between the merged final predictions of the network and the merged final RGB ground truth. We compared these two label maps pixel-wise and generated a multiclass confusion matrix, indicating for each class: the number of pixel correctly identified belonging to that class, the True Positives (TP) and not belonging to it, the True Negatives (TN); and the number of pixels wrongly identified belonging to that class, the False Positives (FP), and not belonging to it, the False Negatives (FN).

Finally, the TP, TN, FP and FN values are used to calculate the *accuracy*, *precision*, *recall* and *F1 score* for each class. Indicating how well each class was classified after merging the gray scale results and assigning the higher confidence one class to each pixel. Figure 3 represent the evaluation workflow.

4. Experimental results and discussion

This section presents and discusses the results of the neural network classification evaluation when trained and tested using each one of the three sets of classes established in Subsection 3.2.

Table 2
Set 1 confusion matrix.

Real \ Predicted	Po	Rock	Sand	Back
Po	94.0	3.8	1.6	0.6
Rock	1.2	97.4	1.3	0.1
Sand	0.4	3.4	95.7	0.6
Back	54.4	28.1	1.2	16.3

Table 3
Set 1 results.

	Area	Acc.	Prec.	Rec.	F1	AUC
Po	50,1	95,4	93,6	97,4	95,5	96,6
Rock	38,4	93,8	90,3	94,0	92,1	98,6
Sand	5,5	98,4	79,3	95,7	86,7	99,6
Back	6,0	94,7	77,7	16,3	27,0	84,4

For each set, we present two tables. The first one showcases the obtained multiclass confusion matrix. The second table presents the AUC value obtained for each class evaluated individually (first evaluation step), along with the *Accuracy*, *Precision*, *Recall* and *F1 score* for each class obtained from the merged final label map evaluation (second evaluation step). Furthermore, in order to give some prospective of the importance of each class, we added a column in the second table indicating the percentage of area present in the ground truth corresponding to each class.

4.1. Set 1 results

Tables 2 and 3 present the results obtained from training and testing the network using the Set 1 classes.

From the multiclass confusion matrix in Table 2, it can be seen that the Po, Rock and Sand classes are identified very accurately. On the other hand the Back class is mostly missclassified as Po and Rock. This results can also be seen looking at the results in Table 3, where the Po, Rock and Sand classes achieve relatively high *F1 scores*, while the Back class only reaches a 27%.

Also, it can be seen in Table 3 that the AUC values are high for all classes, reaching numbers around 98% for the Po, Rock and Sand classes, and a value of 84.4% for the Back class. Following the criteria established in [14], these AUC values represent excellent and good classifiers, respectively. That indicates that the network is able to identify each class when evaluated independently, but as the predictions are merged into the final prediction, the Back class is mostly overlapped by the others, due to its lower confidence levels. The network is not able to classify the Back class with high confidence levels due to the fact of being so diverse, as it contains all the elements that could not be classified into the other classes, making harder for the network to extract generalised features.

Figure 4 shows a qualitative comparison between the semantic network prediction and the ground truth label maps of two test images when the network was trained and tested using the Set 1 classes.

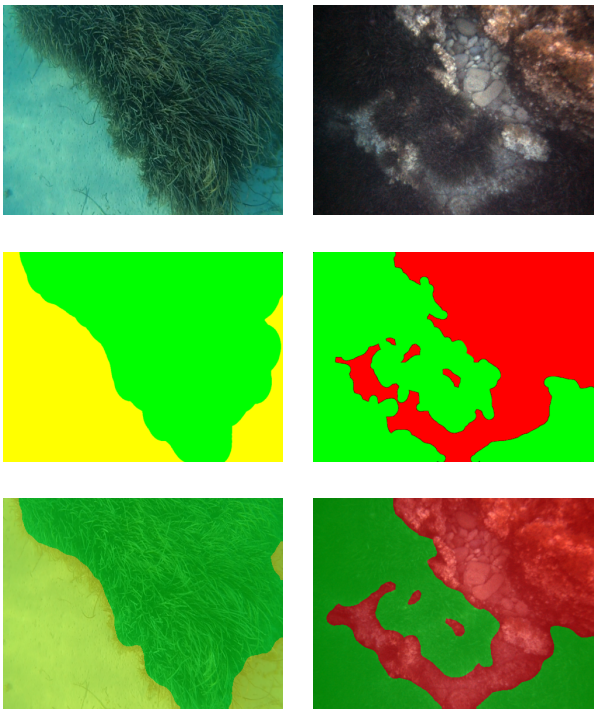


Figure 4: Set 1 prediction examples. First row: Raw input image. Second row: ground truth image. Third row: semantic segmentation output superimposed to the original images.

Table 4
Set 2 confusion matrix.

Real \ Predicted	Po-a	Po-d	Rock	Sand	Back
Po-a	98.5	0.0	1.2	0.3	0.0
Po-d	39.6	19.1	19.8	21.6	0.0
Rock	3.8	0.0	96.1	0.1	0.0
Sand	3.1	5.1	9.9	81.9	0.1
Back	29.4	0.0	62.6	0.4	7.7

Table 5
Set 2 results.

	Area	Acc.	Prec.	Rec.	F1	AUC
Po-a	49,7	95,7	93,3	98,5	95,8	98,8
Po-d	0,4	99,4	19,9	19,1	19,5	86,1
Rock	38,4	93,5	88,1	96,1	91,9	96,8
Sand	5,5	98,7	93,9	81,9	87,5	98,8
Back	6,0	94,4	94,9	7,7	14,3	84,0

4.2. Set 2 results

Tables 4 and 5 present the results obtained from training and testing the network using the Set 2 classes.

From the confusion matrix it can be seen that the Po-a, Rock and Sand classes were classified very accurately, while the Back class is, again, mostly missclassified as Po-a and Rock. The new added class in Set 2, the Po-d (areas with congregations of dead *P. oceanica shoots*), showed poor results, as it was only classified correctly in 34.9% of the cases and was missclassified as the Po-a class 17.2% of the

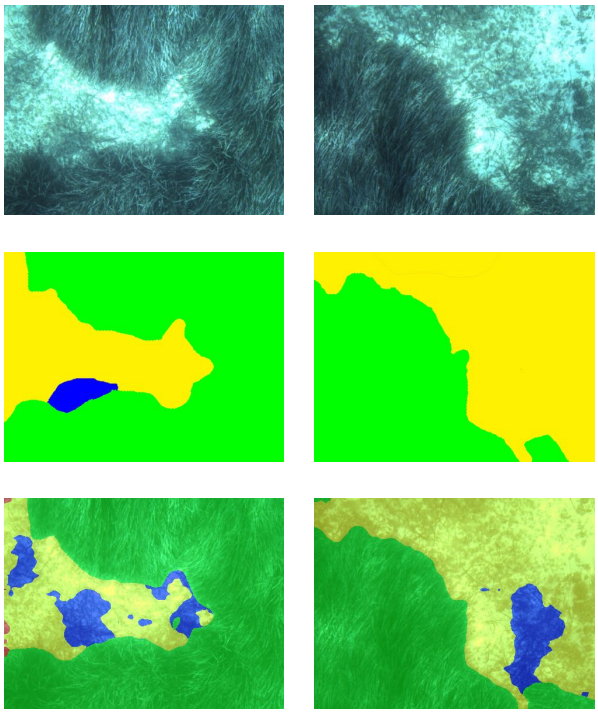


Figure 5: Set 2 prediction examples. First row: Raw input image. Second row: ground truth image. Third row: semantic segmentation output superimposed to the original images.

times , due to the high similarity with this class, and 38.2% of the times as the Sand class, due to the fact that the dead shoots of *P. oceanica* are mostly found lying over sandy areas. This can also be appreciated by looking at the *F1 scores* on Table 5, where the values of the Po-a, Rock and Sand classes are very high, while the ones presented by the Po-d and Back are 19.1% and 7.7%, respectively.

On Table 5 it can also be seen that the percentage of area corresponding to the Po-d class is only 0.4%, meaning that the network almost did not have information corresponding to this class to train on. On the other hand, the percentage of area corresponding to the visually similar Po-a class is much higher, a 49.7%, allowing the network to extensively train on this class and to identify it easily, adding to the reasons why the Po-d is largely missclassified as Po-a. Finally, the obtained AUC values were high for all classes when evaluated individually. The Po-a, Rock and Sand classes fall into the excellent rating and the Po-d and Back into the good one.

Figure 5 shows a qualitative comparison between the semantic network prediction and the ground truth label maps of two test images when the network was trained and tested using the Set 2 classes.

4.3. Set 3 results

Tables 6 and 7 present the results obtained from training and testing the network using the Set 3 classes.

The results obtained for the Po-a, Po-d, Rock, Sand, and Back classes are similar to those obtained for Set 1 and 2. The Po-a, Rock and Sand classes obtained high classification

Table 6
Set 3 confusion matrix.

		Predicted					
		Po-a	Po-d	Rock	Sand	Matte	Back
Real	Po-a	95.8	0.0	2.5	1.5	0.0	0.2
	Po-d	17.2	34.9	9.1	38.2	0.0	0.6
	Rock	2.0	0.0	92.7	4.3	0.0	1.0
	Sand	0.8	5.5	1.1	92.2	0.4	0.1
	Matte	6.8	33.3	0.0	54.2	5.7	0.0
	Back	23.5	0.0	52.6	2.0	0.0	21.9

Table 7
Set 3 results.

	Area	Acc.	Prec.	Rec.	F1	AUC
Po-a	49.7	95.6	95.3	95.8	95.5	98.6
Po-d	0.4	99.2	19.6	34.9	25.1	97.2
Rock	38.3	92.7	88.7	92.7	90.7	97.7
Sand	4.8	96.6	59.4	92.2	72.3	98.9
Matte	0.7	99.3	70.1	5,7	10.6	97.4
Back	6.0	94.8	73.0	21.9	33.8	88.2

results, with *F1 score* values of 95.5%, 92.7% and 72.3%, respectively. In this case, the Sand class performance was lower than the previous sets since the number of missclassified areas of other classes into the Sand class was higher. The Po-d and Back classes only were correctly identified in a 34.9% and 21.9% of the cases, respectively. Additionally, all those classes presented excellent results when evaluated individually, presenting AUC values around 97.5%.

Finally, the new added class in Set 3, the Matte (*P. oceanica* dead matte areas), was only correctly classified 5.7%. It was mostly missclassified as Po-d and Sand. The reasons for that include the fact that, as occurred for the Po-d class, only a 0.7% of the total area corresponded to the Matte class, making hard for the network to train on it. Also, it is worth mentioning that, during the manual ground truth labelling process, it is sometimes hard to distinguish clearly the Matte class, as it is usually partially covered by dead shoots of *P. oceanica* or sand. This issue also contributed in the poor performance over the Matte class. Nonetheless, an AUC value of 88.2% was achieved for the Matte class when evaluated individually, meaning that the network is able to identify it, but with low confidence percentages.

Figure 6 shows a qualitative comparison between the semantic network prediction and the ground truth label maps of two test images when the network was trained and tested using the Set 3 classes.

5. Conclusions and future work

This paper presented the usage of a semantic segmentation deep neural neural network architecture to perform a classification of *P. oceanica* meadows and its marine seafloor habitat. The main advantage of the system presented versus the current state of the art techniques are that: it is able to perform a pixel-wise segmentation, not losing any information, without needing any post-processing; and the

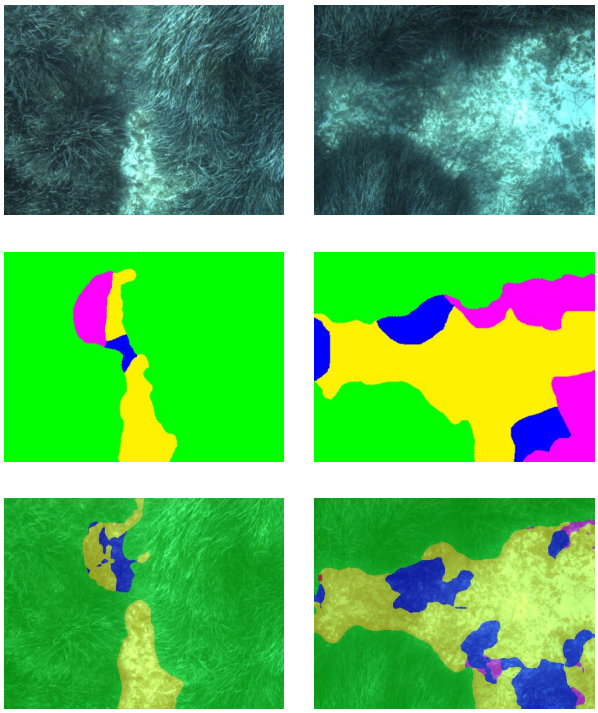


Figure 6: Set 3 prediction examples. First row: Raw input image. Second row: ground truth image. Third row: semantic segmentation output superimposed to the original images.

fact that it is able to distinguish between several classes at the same time.

The neural network evaluation over three different sets of classes showed very high performance metrics for the Po-a, Rock and Sand classes, which are three most area-wise dominant classes, being as well the most important for the research objective. On Sets 2 and 3, the PO-d and Matte classes were introduced, trying to define better the seafloor habitat and *P. oceanica* status, obtaining more detailed information. Nonetheless, the results showed poor classification performance for those classes in the final predictions, although it was able to identify them when evaluated separately, before being overlapped by higher confidence classes afterwards.

The poor performance obtained over these classes can be mainly attributed to two factors. The first one is the fact that these classes represent a small percentage of area of the *P. oceanica* seafloor habitat with respect to other classes like the Po-a or Rock, making it hard for the network to learn its features and consequently, being able to classify them correctly. The network always tends to classify these small represented classes into one of the more abundant ones. The second fact that affected the performance of these classes is the high similarity between some of them, making it hard to be distinguished even during the manual ground truth labelling process. For example, there are areas where it is hard to determine if the *P. oceanica* present is alive (Po-a class) or it is a voluminous congregation of dead shoots (Po-d class). Also, it is hard to establish a strict classification when some of the classes are overlapped, such as dead shoots of *P. oceanica* laid over a rock or sand; or sand being on top of a *P. oceanica* dead matte.

Further developments will focus on the acquisition of new data containing seafloor habitat elements of the less represented classes (Po-d and Matte) in order to train over more data and learn its features better. Also, new hyperparameter setting of the network training could be tested.

As of the work presented in this paper, the final merged prediction for all classes is generated by selecting the class with higher confidence for each pixel. Seeing the performance drop that the network had when classifying lower area-wise represented classes between the individual evaluation versus the merged one. Some rules could be applied to that final merged prediction generation process. The goal would be to favour the appearance of these inferior represented classes even when the confidence level was lower.

Finally, in order to help distinguish between visually similar classes live Po-a from Po-d, we will work in the introduction of 3D information data into the training process, as it is available since the images were acquired using a stereo camera. This will help the network to extract features based on the 3D information, being able to better differentiate classes like healthy voluminous *P. oceanica* meadows (Po-a) from congregations of dead shoots lying on the seafloor (Po-d).

Acknowledgments

This work is partially supported by Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contracts TIN2014-58662-R and DPI2017-86372-C3-3-R.

CRediT authorship contribution statement

Miguel Martin-Abadal: Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. **Ivan Riutort-Ozcariz:** Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. **Gabriel Oliver-Codina:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Yolanda Gonzalez-Cid:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

[1] Alagna, A., Fernández, T.V., Anna, G.D., Magliola, C., Mazzola, S., Badalamenti, F., 2015. Assessing posidonia oceanica seedling substrate preference: An experimental determination of seedling anchorage success in rocky vs. sandy substrates. PLOS ONE 10, 1–15. URL: https://doi.org/10.1371/journal.pone.0125321, doi:doi:10.1371/journal.pone.0125321.

[2] Bonin-Font, F., Burguera, A., Lisani, J.L., 2017. Visual discrimination and large area mapping of posidonia oceanica using a lightweight auv. IEEE Access 5, 24479–24494.

[3] Caughlan, L., 2001. Cost considerations for long-term ecological monitoring. Ecological Indicators 1, 123–134.

[4] Del Vecchio, S., Fantinato, E., Silan, G., Buffa, G., 2018. Trade-offs between sampling effort and data quality in habitat monitoring. Biodiversity and Conservation 28, 55–73.

[5] Diaz-Almela, E., Duarte, C., 2008. Management of Natura 2000 Habitats 1120, (Posidonia Oceanicae). Technical Report. European Commission.

[6] Dumoulin, V., Visin, F., 2016. A guide to convolution arithmetic for deep learning. ArXiv e-prints - 1603.07285 arXiv:1603.07285.

[7] Gonzalez-Cid, Y., Burguera, A., Bonin-Font, F., Matamoros, A., 2017. Machine learning and deep learning strategies to identify posidonia meadows in underwater images. OCEANS 2017 - Aberdeen , 1–5.

[8] Guerrero-Meseguer, L., Sanz-Lázaro, C., Suk-ueng, K., Marín, A., 2016. Influence of substrate and burial on the development of posidonia oceanica: Implications for restoration. Restoration Ecology doi:doi: 10.1111/rec.12438.

[9] Kiparissis, S., Fakiris, E., Papatheodorou, G., Geraga, M., Kornaros, M., Kapareliotis, A., Ferentinos, G., 2011. Illegal trawling and induced invasive algal spread as collaborative factors in a posidonia oceanica meadow degradation. Biological Invasions 13, 669–678. URL: http://dx.doi.org/10.1007/s10530-010-9858-9, doi:doi: 10.1007/s10530-010-9858-9.

[10] Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 3431–3440doi:doi: 10.1109/CVPR.2015.7298965.

[11] Marba, N., Duarte, C., 2010. Mediterranean warming triggers seagrass (posidonia oceanica) shoot mortality. Global Change Biology 16, 2366–2375.

[12] Martin-Abadal, M., Guerrero-Font, E., Bonin-Font, F., Gonzalez-Cid, Y., 2018. Deep semantic segmentation in an auv for online posidonia oceanica meadows identification. IEEE Access 6, 60956–60967. doi:doi: 10.1109/ACCESS.2018.2875412.

[13] Milazzo, M., Badalamenti, F., Ceccherelli, G., Chemello, R., 2004. Boat anchoring on posidonia oceanica beds in a marine protected area (italy, western mediterranean): Effect of anchor types in different anchoring stages. Journal of Experimental Marine Biology and Ecology 299, 51–62. doi:doi: 10.1016/j.jembe.2003.09.003.

[14] Powers, D.M.W., 2011. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. International Journal of Machine Learning Technology 2, 37–63.

[15] Provost, F., Fawcett, T., Kohavi, R., 2001. The case against accuracy estimation for comparing induction algorithms. Proceedings of the Fifteenth International Conference on Machine Learning .

[16] Lopez y Royo, C., Pergent, G., Pergent-Martini, C., Casazza, G., 2010. Seagrass (posidonia oceanica) monitoring in western mediterranean: Implications for management and conservation. Environmental monitoring and assessment 171, 365–80.

[17] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. ArXiv e-prints - 1409.1556 arXiv:1409.1556.

[18] Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., Urtasun, R., 2016. MultiNet: Real-time joint semantic reasoning for autonomous driving. ArXiv e-prints - 1612.07695 arXiv:1612.07695.

[19] Telesca, L., Belluscio, A., Criscoli, A., Ardizzone, G., Apostolaki, E.T., Fraschetti, S., Gristina, M., Knittweis, L., Martin, C.S., Pergent, G., Alagna, A., Badalamenti, F., Garofalo, G., Gerakaris, V., Pace, M.L., Pergent-Martini, C., Salomidi, M., 2015. Seagrass meadows (posidonia oceanica) distribution and trajectories of change. Scientific reports .

Chapter 3

Pipeline characterisation

This chapter presents the work carried out on underwater pipe and valve recognition and characterisation.

Over the last few decades, underwater intervention has experienced an uprise due to the increasing need to perform inspection and intervention tasks on industrial infrastructures, such as offshore oil and gas rigs or underwater pipeline networks (Yu et al., 2017; Jacobi and Karimanzira, 2013).

Recently, the usage of Autonomous Underwater Vehicles and manipulators has eased the workload and risks of such interventions, automating these tasks by gathering information from their surroundings, interpreting it and making decisions based on it (Ridao et al., 2015; Heshmati-Alamdari et al., 2018).

The objective of this work is to design an automated system that can identify and gather information on valves, pipes, and structural elements of underwater pipeline networks. Later, the different elements should be positioned in a 3D space to provide information during manipulation tasks and build information maps to accurately depict the layout of a pipeline network.

The first step was to collect point cloud data to train and test a 3D deep learning segmentation architecture. Several hundred point clouds, containing different layouts of underwater pipes and valves, were generated using stereoscopic vision from a pair of cameras mounted on diverse marine vehicles. Following, two deep learning architectures were implemented and tested to find the best performing hyperparameters for pipe and valve segmentation. Finally, algorithms were developed to extract manipulation information from the detected instances, such as pipe vectors, gripping points, the position of structural elements like elbows or connections, and valve type and orientation. Additionally, if point clouds are spatially referenced, an information map of an inspected area can be created.

All work is described in detail in two published journal papers. The first one details the data gathering process and the network selection, training, and evaluation, as well as hyperparameter study in terms of segmentation performance and computational time. The second article presents an upgrade of the used segmentation network and introduces new training and testing data. Additionally, the information extraction and unification algorithms are described and validated. Finally, the article describes the online implementation and execution of the network and algorithms on an AUV, providing real-time information for inspection and manipulation tasks.

Title: Underwater Pipe and Valve 3D Recognition Using Deep Learning Segmentation
Authors: **M. Martin-Abadal**, M. Piñar-Molina, A. Martorell-Torres, G. Oliver-Codina and Y. Gonzalez-Cid
Journal: Journal of Marine Science and Engineering
Published: 23 December 2020
Quality index: JCR2021 *Engineering, marine*, IF 2.744, Q1 (4/16)

Title: Real-time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks
Authors: **M. Martin-Abadal**, G. Oliver-Codina and Y. Gonzalez-Cid
Journal: Sensors
Published: 24 October 2022
Quality index: JCR2021 *Engineering, electrical & electronic*, IF 3.847, Q2 (95/276)

Underwater Pipe and Valve 3D Recognition Using Deep Learning Segmentation

Miguel Martin-Abadal*, Manuel Piñar-Molina, Antoni Martorell-Torres, Gabriel Oliver-Codina and Yolanda Gonzalez-Cid

Department of Mathematics and Computer Science. University of the Balearic Islands, 07122, Palma, Spain

ARTICLE INFO

The work presented in this preprint has been published in the *Journal of Marine Science and Engineering* as:

Martin-Abadal, M.; Piñar-Molina, M.; Martorell-Torres, A.; Oliver-Codina, G.; Gonzalez-Cid, Y. *Underwater Pipe and Valve 3D Recognition Using Deep Learning Segmentation.* J. Mar. Sci. Eng. 2021, 9, 5.
DOI: 10.3390/jmse9010005

ABSTRACT

During the past few decades, the need to intervene in underwater scenarios has grown due to the increasing necessity to perform tasks like underwater infrastructure inspection and maintenance or archaeology and geology exploration. In the last few years, the usage of Autonomous Underwater Vehicles (AUVs) has eased the workload and risks of such interventions. To automate these tasks, the AUVs have to gather the information of their surroundings, interpret it and make decisions based on it. The two main perception modalities used at close range are laser and video. In this paper, we propose the usage of a deep neural network to recognise pipes and valves in multiple underwater scenarios, using 3D RGB point cloud information provided by a stereo camera. We generate a diverse and rich dataset for the network training and testing, assessing the effect of a broad selection of hyperparameters and values. Results show *F1-scores* of up to 97.2% for a test set containing images with similar characteristics to the training set and up to 89.3% for a secondary test set containing images taken at different environments and with distinct characteristics from the training set. This work demonstrates the validity and robust training of the PointNet neural in underwater scenarios and its applicability for AUV intervention tasks.

1. Introduction

During the past few decades, the interest in underwater intervention has grown exponentially as more often it is necessary to perform underwater tasks like surveying, sampling, archaeology exploration or industrial infrastructure inspection and maintenance of offshore oil and gas structures, submerged oil wells or pipeline networks, among others [2, 7, 11, 24, 52].

Historically, scuba diving has been the prevailing method of conducting the aforementioned tasks. However, performing these missions in a harsh environment like open water scenarios is slow, dangerous, and resource consuming. More recently, thanks to technological advances such as Remotely Operated Vehicles (*ROVs*) equipped with manipulators, more deep and complex underwater scenarios are accessible for scientific and industrial activities.

Nonetheless, these *ROVs* have complex dynamics that make their piloting a difficult and error-prone task, requiring trained operators. In addition, these vehicles require a support vessel, which leads to expensive operational costs. To mitigate that, some research centres have started working towards intervention Autonomous Underwater Vehicles (*AUVs*) [20, 35, 48]. In addition, due to the complexity of the Underwater Vehicle Manipulator Systems (UVMS), recent studies have been published towards its control [19, 39].

Traditionally, when operating in unknown underwater environments, acoustic bathymetric maps are used to get a first identification of the environment. Once the bathymetric information is available, *ROVs* or *AUVs* can be sent to obtain more detailed information using short distance sensors with higher resolution. The two main perception modalities used at close range are laser and video, thanks to their high resolution. They are used during the approach, object recognition and intervention phases. Existing solutions for all perception modalities are reviewed in Section 2.1.

The underwater environment is one of the most problematic in terms of sensing in general and in terms of object perception in particular. The main challenges of underwater perception include distortion in signals, light propagation artefacts like absorption and scattering, water turbidity changes or depth-depending colour distortion.

Accurate and robust object detection, identification of target objects in different experimental conditions and pose estimation are essential requirements for the execution of manipulation tasks.

In this work, we propose a deep learning based approach to recognise pipes and valves in multiple underwater scenarios , using the 3D RGB point cloud information provided by a stereo camera, for real-time *AUV* inspection and manipulation tasks.

The remainder of this paper is structured as follows: Section 2 reviews related work on underwater perception and pipe and valve identification and highlights the main contributions of this work. Section 3 describes the adopted methodology and materials used in this study. The experimental results are presented and discussed in Section 4. Finally, Section 5 outlines the main conclusions and future work.

2. Related Work and Contributions

2.1. State of the Art

Even though computer vision is one of the most complete and used perception modalities in robotics and object

*Corresponding author
miguel.martin@uib.es (M. Martin-Abadal)

recognition tasks, it has not been widely used in underwater scenarios. Light transmission problems and water turbidity affect the images clarity, colouring and produce distortions; these factors have favoured the usage of other perception techniques.

Sonar sensing has been largely used for object localisation or environment identification in underwater scenarios [4, 27]. In [29], Kim et al. present an AdaBoost based method for underwater object detection, while Wang et al. [51] propose a combination of non-local spatial information and frog leaping algorithm to detect underwater objects in sonar images. More recently, object detection deep learning techniques have started to apply over sonar imaging in applications such as detection of underwater bodies in [33, 34] or underwater mine detection in [12]. Sonar imaging also presents some drawbacks as it tends to generate noisy images, losing texture information; and are not capable of gathering colour information, which is useful in object recognition tasks.

Underwater laser scans are another perception technique used for object recognition, providing accurate 3D data. In [43], Palomer et al. present the calibration and integration of a laser scanner on an AUV for object manipulation. Himri et al. [21, 22] use the same system to detect objects using a recognition and pose estimation pipeline based on point cloud matching. Inzartsev et al. [23] simulate the use of a single beam laser paired with a camera to capture its deformation and track an underwater pipeline. Laser scans are also affected by light transmission problems, have a very high initial cost and can only provide colourless point clouds.

The only perception modality that allows gathering of colour information for the scene is computer vision. Furthermore, some of its aforementioned weaknesses can be mitigated by adapting to the environmental conditions, adjusting the operation range, calibrating the cameras or colour correcting the obtained images.

Traditional computer vision approaches have been used to detect and track submerged artifacts [1, 9, 41, 44], cables [13, 38, 42] and even pipelines [15, 38, 50, 53]. Some works are based on shape and texture descriptors [15, 38] or template matching [30, 32], while others exploit colour segmentation to find regions of interest in the images, which are later further processed [3, 44].

On pipeline detection, Kallasi et al. in [28] and Razzini et al. in [35, 36] present traditional computer vision methods combining shape and colouring information to detect pipes in underwater scenarios and later project them into point clouds obtained from stereo vision. In these works, the point cloud information is not used to assist the pipe recognition process.

The first found trainable system to detect pipelines is presented in [47] by Rekik et al. using the objects structure and content features along a Support Vector Machine to classify between positive and negative underwater pipe images samples. Later, Nunes et. al introduced the application of a Convolutional Neural Network in [40] to classify up to five underwater objects, including a pipeline. In both of these works, no position of the object is given, but simply a binary output on the object's presence.

The application of computer vision approaches based on deep learning in underwater scenarios has been limited to the detection and pose estimation of 3D-printed objects in [26] or for living organisms detection like fishes [25] or jellyfishes [37]. Few research studies involving pipelines are restricted to damage evaluation [10, 31] or valve detection for navigation [46] working with images taken from inside the pipelines. The only known work addressing pipeline recognition using deep learning is from Guerra et al. in [17], where a camera-equipped drone is used to detect pipelines in industrial environments.

To the best knowledge of the authors, there are not works applying deep learning techniques in underwater computer vision pipeline and valve recognition, nor implementing the usage of point cloud information on the detection process itself.

2.2. Main Contributions

The main contributions of this paper are composed of:

1. Generation of a novel point cloud dataset containing pipes and different types of valves in varied underwater scenarios, providing enough data to perform a robust training and testing of the selected deep neural network.
2. Implementation and testing of the PointNet architecture in underwater environments to detect pipes and valves.
3. Studying the suitability of the PointNet network on real-time autonomous underwater recognition tasks in terms of detection performance and inference time by tuning diverse hyperparameter values.
4. The datasets (point clouds and corresponding ground truths) along with a trained model are provided to the scientific community.

3. Materials and Methods

This section presents an overview of the selected network; explains the acquisition, labelling and organisation of the data; and details the studied network hyperparameters, the validation process and the evaluation metrics.

3.1. Deep Learning Network

To perform the pipe and valve 3D recognition from point cloud segmentation, we selected the PointNet deep neural network [8]. This is a unified architecture for applications ranging from object classification and part segmentation to scene semantic segmentation. PointNet is a highly efficient and effective network, obtaining great metrics in both object classification and segmentation tasks in indoor and outdoor scenarios [8]. However, it has never been tested in underwater scenarios. The whole PointNet architecture is shown in Figure 1.

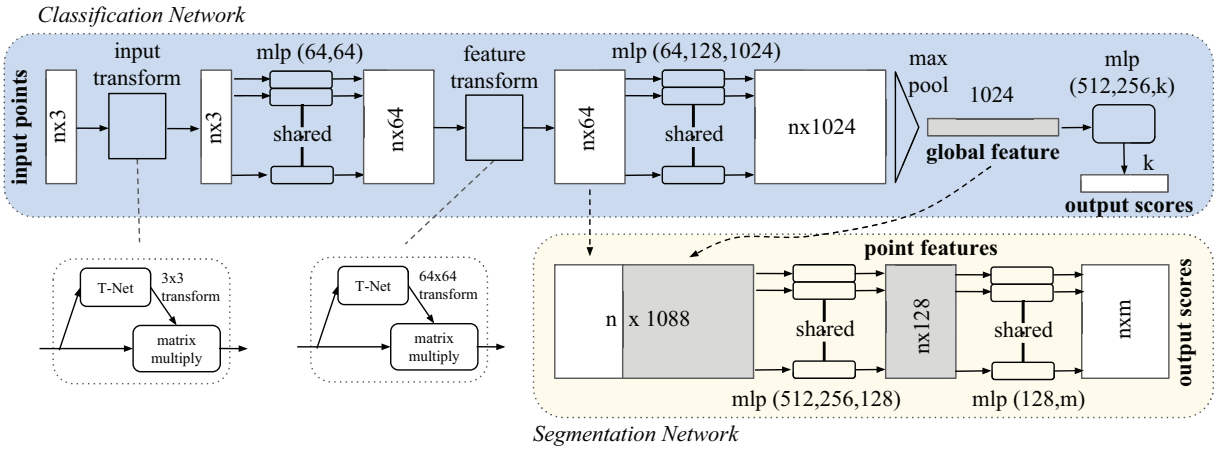


Figure 1: PointNet architecture. Reproduced from [8], with permission from publisher Hao Su, 2020.

In this paper, we use the *Segmentation Network* of PointNet. This network is an extension to the *Classification Network*, as it can be seen in Figure 1. Some of its key features include:

- The integration of max pooling layers as symmetric function to aggregate the information from each point, making the model invariant to input permutations.
- Being able to predict per point features that rely both on local structures from nearby points and global information which makes the prediction invariant to object transformations such as translations or rotations. This combination of local and global information is obtained by concatenating the global point cloud feature vector with the local per point features.
- Making the semantic labeling of a point cloud invariant to the point cloud geometric transformations by aligning all input set to a canonical space before feature extraction. To achieve this, an affine transformation matrix is predicted using a mini-network (T-net in Figure 1) and directly applied to the coordinates of input points.

The PointNet architecture takes as input point clouds and it outputs a class label for each point. During the training, the network is also fed with ground truth point clouds, where each point is labelled with its pertaining class. The labelling process is further detailed in Section 3.2.2.

As the original PointNet implementation, we used a softmax cross-entropy loss along an Adam optimiser. The decay rate for batch normalisation starts with 0.5 and is gradually increased to 0.99. In addition, we applied a dropout with keep ratio 0.7 on the last fully connected layer, before class score prediction. Other hyperparameters values such as learning rate or batch size are discussed, along other parameters, on Section 3.3.

Furthermore, to improve the network performance, we implemented an early stopping strategy based on the work of Prechelt in [45], assuring that the network training process stops at an epoch that ensures minimum divergence between validation and training losses. This technique allows for obtaining a more general and broad training, avoiding overfitting.

3.2. Data

This subsection explains the acquisition, labelling and organisation of the data used to train and test the PointNet neural network.

3.2.1. Acquisition

As mentioned in Section 3.1, the PointNet uses point-clouds for its training and inference. To obtain the point clouds, we set up a Bumblebee2 Firewire stereo rig [14] on an Autonomous Surface Vehicle (ASV) through a *Robot Operating System* (ROS) framework.

First, we calibrated the stereo rig both on fresh and salt water using the ROS package *image_pipeline/camera_calibration* [5, 6]. It uses a chessboard pattern to obtain the camera, rectification and projection matrices along the distortion coefficients for both cameras.

The acquired synchronised pairs of left-right images (resolution: 1024×768 pixels) are processed as follows by the *image_pipeline/stere_image_proc* ROS package [49] to calculate the disparity between pairs of images based on epipolar matching [18], obtaining the corresponding depth of each pixel from the stereo rig.

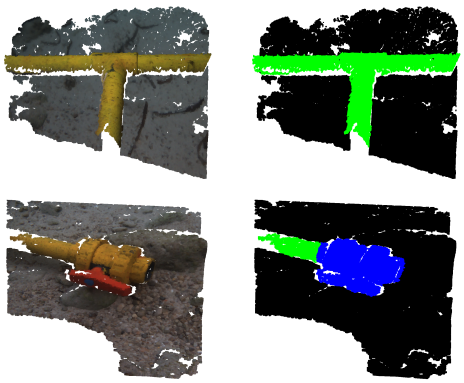
Finally, combining this depth information with the RGB colouring from the original images, we generate the point clouds. An example of the acquisition is pictured in Figure 2

3.2.2. Ground Truth Labelling

Ground truth annotations are manually built from the point clouds, where the pixels corresponding to each class are marked with a different label. The studied classes and their RGB labels are: *Pipe* (0, 255, 0), *Valve* (0, 0, 255) and

(a) (b) (c)

Figure 2: Data acquisition process. (**a**) left and right stereo images, (**b**) disparity image, (**c**) point cloud.



(a) (b)

Figure 3: (**a**) Original point cloud; (**b**) ground truth annotations, points corresponding to pipes are marked in green; to valves, in blue; and to background, in black.

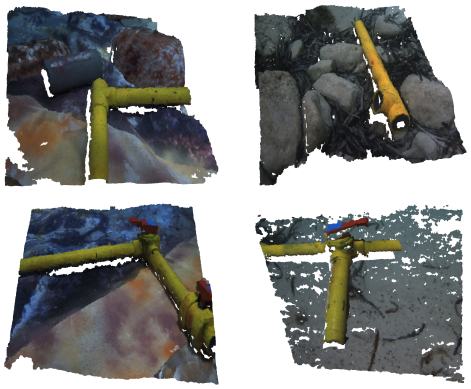
Background $(0, 0, 0)$. Figure 3 shows a couple of point clouds along with their corresponding ground truth annotations.

3.2.3. Dataset Managing

Following the steps described in the previous section, we generated two datasets. The first one includes a total of 262 point clouds along with their ground truths. It was obtained on an artificial pool and contains diverse connections between pipes of different diameters and 2/3 way valves. It also contains other objects such as cement blocks and ceramic vessels, always over a plastic sheeting simulating different textures. This dataset is split into a train-validation set (90% of the data, 236 point clouds) and a test set (10% of the data, 26 point clouds). The different combinations of elements and textures increase its diversity, helping to assure the robustness in the training and reduce overfitting. From now on, we will refer to this dataset as the *Pool* dataset.

The second dataset includes a total of 22 point clouds and their corresponding ground truths. It was obtained in the sea and contains different pipe connections and valves positions. In addition, these 22 point clouds were obtained over diverse types of seabed, such as sand, rocks, algae, or a combination of them. This dataset is used to perform a secondary test, as it contains point clouds with different characteristics of the ones used to train and validate the network, allowing us to assess how well the network generalises its training to new conditions. From now on, we will refer to this dataset as the *Sea* dataset.

Figure 4: Dataset managing.



(a) (b)

Figure 5: Examples of point clouds from (**a**) *Pool* dataset and (**b**) *Sea* dataset.

Figure 4 illustrates the dataset managing, while in Figure 5 some examples of point clouds from both datasets are shown.

3.3. Hyperparameter Study

When training a neural network, there are hyperparameters which can be tuned, changing some of the features of the network or the training process itself. We selected some of these hyperparameters and trained the network using different values to study their effect over its performance in underwater scenarios. The considered hyperparameters were:

- Batch size: number of training samples utilised in one iteration before backpropagating.
- Learning rate: affects the size of the matrix changes that the network takes when searching for an optimal solution.
- Block (B) and stride (S) size: to prepare the network input, the point clouds are sampled into blocks of BxB meters, with a sliding window of stride S meters.
- Number of points: maximum number of allowed points per block. If it exceeds, random points are deleted. Used to control the point cloud density.

The tested values for each hyperparameter are shown in Table 1. In total, 13 experiments are conducted, one using the hyperparameter values used in the original PointNet implementation [8] (marked in bold in Table 1); and 12 more, each one fixing three of the aforementioned hyperparameters to their original values and using one of the other tested values for the fourth hyperparameter. This way, the effect of each hyperparameter and its value over the performance is isolated.

Table 1
Tested hyperparameter values. Original values are marked in bold.

Hyperparameter	Tested values					
Batch size	16	24	**32**			
Learning rate	0.005	**0.001**	0.0002			
Block-stride	2-2	2-1	**1-1**	1-0.75		
Num. points	**4096**	2048	1024	512	256	128

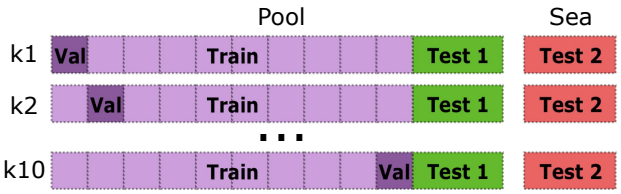


Figure 6: Implementation of the 10k-fold cross-validation method.

Table 2
Pool test set *F1-scores*.

Experiment	F1_Pipe	F1_Valve	F1_Background	F1_Mean
Base	97.0%	93.1%	99.8%	96.6%
Batch 24	96.8%	92.7%	99.8%	96.4%
Batch 16	96.7%	92.3%	99.8%	96.2%
Lr 0005	96.4%	91.0%	99.7%	95.7%
Lr 00002	96.5%	92.5%	99.7%	96.2%
BS 2_2	96.0%	90.8%	99.7%	95.5%
BS 2_1	96.9%	93.3%	**99.8%**	96.7%
BS 1_075	**97.1%**	**94.9%**	99.7%	**97.2%**
Np 2048	96.7%	92.2%	99.8%	96.2%
Np 1024	96.9%	93.2%	99.8%	96.6%
Np 512	96.8%	92.6%	99.8%	96.4%
Np 256	96.9%	93.4%	99.8%	96.7%
Np 128	96.7%	92.8%	99.8%	96.4%

3.4. Validation

3.4.1. Validation Process

To ensure the robustness of the results generated for the 13 experiments, we used the 10 k-fold cross-validation method [16]. Using this method, the train-validation set of the *Pool* dataset is split into ten equally sized subsets. The network is trained ten times as follows, each one using a different subset as validation (23 point clouds) and the nine remaining as training (213 point clouds), generating ten models which are tested against both *Pool* and *Sea* test sets. Finally, each experiment performance is computed as the mean of the results of its 10 cross-validation models. This method reduces the variability of the results, as these are less dependent on the selected training and validation subsets, therefore obtaining a more accurate performance estimation. Figure 6 depicts the k-fold cross-validation technique applied to the dataset managing described in Section 3.2.3

3.4.2. Evaluation Metrics

To evaluate a model performance, we make a point-wise comparison between its predictions and their corresponding ground truth annotations, generating a multi-class confusion matrix. This confusion matrix indicates, for each class: the number of points correctly identified belonging to that class, *True Positives* (TP) and not belonging to it, *True Negatives* (TN); the number of points misclassified as the studied class, *False Positives* (FP); and the number of points belonging to that class misclassified as another one, *False Negatives* (FN). Finally, the TP, FP and FN values are used to calculate the *Precision*, *Recall* and *F1-score* for each class, following Equations (1)–(3):

$$Precision = \frac{TP}{TP + FP} \quad , \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \quad , \tag{2}$$

$$F1\text{-}score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad . \tag{3}$$

Additionally, the mean time that a model takes to perform the inference of a point cloud is calculated. This metric is very important, as it defines the frequency that information is provided to the system. In underwater applications, it would directly affect the agility and responsiveness of the AUV that this network could be integrated in, having an impact over the final operation time.

4. Experimental Results and Discussion

This section reports the performance obtained for each experiment over the *Pool* and *Sea* test sets and discusses the effect of each hyperparameter over it. The notation used to name each experiment corresponds as follows: "Base" for the experiment conducted using the original hyperparameter values, marked in bold in Table 1; the other experiments are notated as an abbreviation of the modified hyperparameter for that experiment ("Batch" for batch size, "Lr" for learning rate, "BS" for block-stride and "Np" for number of points) followed by the actual value of the hyperparameter for that experiment. For instance, experiment *Batch 24* uses all original hyperparameter values except for the batch size, which in this case is 24.

4.1. *Pool* Dataset Results

Table 2 shows the *F1-scores* obtained for the studied classes and its mean for all experiments when evaluated over the *Pool* test set. The mean inference time for each experiment is showcased in Figure 7 as follows.

The results presented in Table 2 show that all experiments achieved a mean *F1-score* greater than 95.5%, with the highest value of 97.2% for the experiment *BS 1_075*, which has a smaller block stride than its size, overlapping information. Considering the figures of mean *F1-score* for all experiments, it is safe to say that no hyperparameter seemed to represent a major shift in the network behaviour.

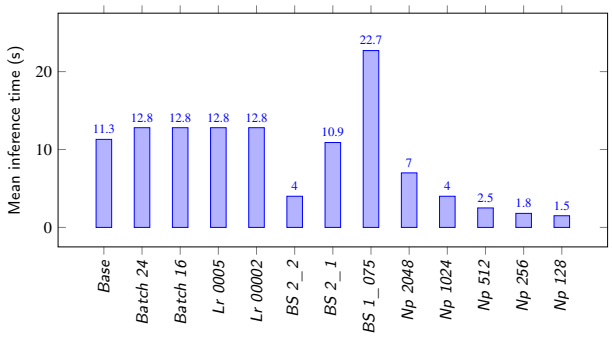
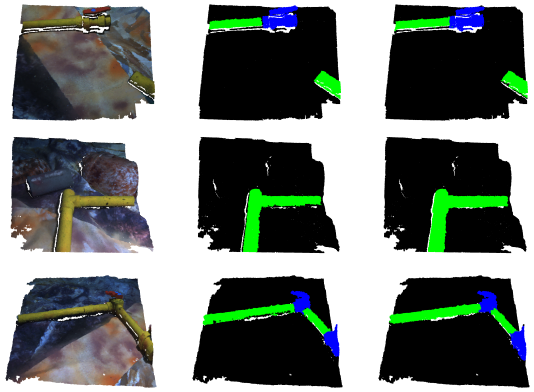


Figure 7: *Pool* test set mean inference time.



(a) (b) (c)

Figure 8: Qualitative results for the *Pool* test set. (**a**) original point cloud, (**b**) ground truth annotations, (**c**) network prediction.

Looking at the metrics presented by the best performing experiment for each class, it can be seen that the *Pipe* class achieved an *F1-score* of 97.1%, outperforming other state-of-the-art methods for underwater pipe segmentation: [28]—traditional computer vision algorithms over 2D underwater images achieving an *F1-score* of 94.1%, [35]—traditional computer vision algorithms over 2D underwater images achieving a mean *F1-score* over three datasets of 88.0% and [17]—deep leaning approach for 2D drone imagery achieving a pixel-wise accuracy of 73.1%. For the valve class, the *BS 1_075* experiment achieved a *F1-score* of 94.9%, being a more challenging class due to its complex geometry. As far as the authors know, no comparable work on underwater valve detection has been identified. Finally, for the more prevailing *Background* class, the best performing experiment achieved an *F1-score* of 99.7%.

The results on mean inference time for each experiment presented in Figure 7 shows that the batch size and learning rate hyperparameter values do not influence the inference time or have little impact, as their value is very similar to the one obtained in the *Base* experiment. On the contrary, the block and stride size highly affect the inference time, the bigger the information block or the stride between blocks, the faster the network can analyse a point cloud, and vice versa. Finally, the maximum number of allowed points per block also has a direct impact over the inference time, the lower it is, the faster the network can analyse a point cloud, as it becomes less dense. The time analysis was carried out in a computer with the following specs—processor: Intel i7-7700, RAM: 16 GB, GPU: NVIDIA GeForce GTX 1080.

Taking into account both metrics, *BS 1_075* presented the best *F1-score* and has the highest inference time. In this experiment, the network uses a small block size and stride, being able to analyse the data and extract its features better, at the cost of taking longer. The hyperparameter values of this experiment are a good fit for a system in which quick responsiveness to changes and high frequency of information are not a priority, allowing for maximising the recognition performance.

On the other hand, experiments such as *BS 2_2* or *Np 1024*, *512*, *256*, *128* were able to maintain very high *F1-scores* while significantly reducing the inference time. The hyperparameter values tested in these experiments are a good fit for more agile systems that need a higher frequency of information and responsiveness to changes.

Table 3
Sea test set *F1-scores*.

Experiment	F1_Pipe	F1_Valve	F1_Background	F1_Mean
Base	85.9%	79.5%	98.8%	88.1%
Batch 24	87.2%	79.9%	98.9%	88.7%
Batch 16	**88.1%**	80.9%	**99.0%**	**89.3%**
Lr 0005	86.2%	**81.2%**	98.8%	88.7%
Lr 00002	85.2%	76.3%	98.7%	86.8%
BS 2_2	80.7%	77.2%	**97.9%**	85.3%
BS 2_1	80.2%	79.7%	97.6%	85.8%
BS 1_075	86.7%	73.9%	99.0%	86.5%
Np 2048	85.2%	80.1%	98.5%	87.9%
Np 1024	86.1%	77.8%	98.8%	87.6%
Np 512	85.4%	70.7%	98.8%	85.0%
Np 256	87.1%	80.2%	98.9%	88.8%
Np 128	84.5%	71.5%	98.7%	84.9%

Figure 8 shows some examples of original point clouds from the *Pool* test set along with their corresponding ground truth annotations and network predictions.

4.2. *Sea* Dataset Results

Table 3 shows the *F1-scores* obtained for the studied classes and its mean, for all experiments when evaluated over the *Sea* test set. The mean inference time for each experiment is showcased in Figure 9 as follows.

The results presented in Table 3 show that all experiments achieved a mean *F1-score* greater than 84.9% with the highest value of 89.3% for the experiment *Batch 16*. On average, the mean *F1-score* was around 9% lower than for the *Pool* test set. Even so, all experiments maintained high *F1-scores*. Again, the *F1-scores* of the *Pipe* and *Valve* classes are relatively lower than for the *Background* class. Even though the *Sea* test set is more challenging, as it contains unseen pipe and valve connections and environment conditions, the network was able to generalise its training and avoid overfitting.

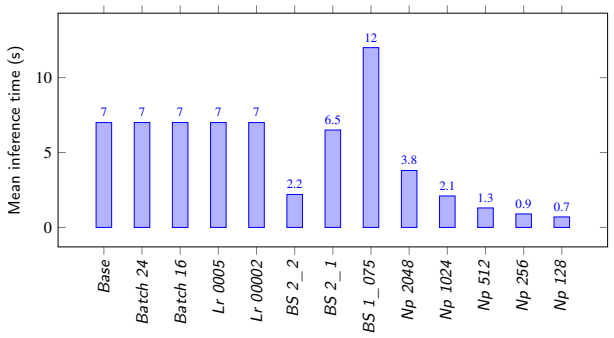


Figure 9: *Sea* test set mean inference time.

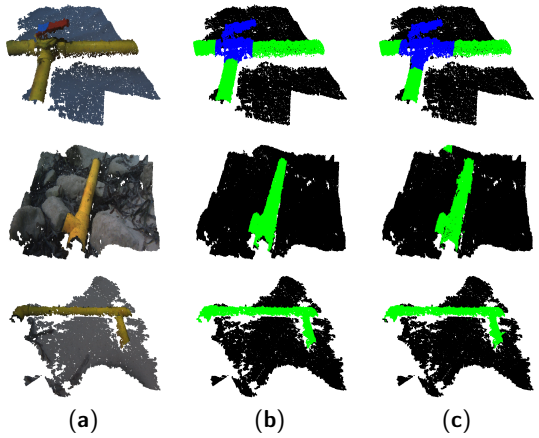


Figure 10: Qualitative results for the *Sea* test set. (**a**) original point cloud; (**b**) ground truth annotations; (**c**) network prediction.

The results on mean inference time for each experiment presented in Figure 9 shows that the mean inference times for the *Sea* test set are proportionally lower than the *Pool* test set for all experiments. This occurs because the *Sea* test set contains smaller point clouds with fewer points.

Figure 10 shows some examples of original point clouds from the *Sea* test set along with their corresponding ground truth annotations and network predictions.

5. Conclusions and Future Work

This work studied the implementation of the PointNet deep neural network in underwater scenarios to recognise pipes and valves from point clouds. First, two datasets of point clouds were gathered, providing enough data for the training and testing of the network. From these, a train-validation set and two test sets were generated, a primary test set with similar characteristics as the training data and a secondary one containing unseen pipe and valve links and environment conditions to test the network training generalisation and overfitting. Then, diverse hyperparameter values were tested to study their effect over the network performance, both in the recognition task and inference time.

Results from the recognition task concluded that the network was able to identify pipes and valves with high accuracy for all experiments in both *Pool* and *Sea* test sets, reaching *F1-scores* of 97.2% and 89.3%, respectively. Regarding the network inference time, results showed that it is highly dependent on the size of information block and its stride; and to the point clouds density.

From the performed experiments, we obtained a range of models covering different trade-offs between detection performance and inference time, enabling the network implementation into a wider spectrum of systems, adapting to its detection and computational cost requirements. The *BS 1_075* experiment presented metrics that fitted a slower, more still system, while experiments like *BS 2_2* or *Np 1024*, *512*, *256*, *128* are a good fit for more agile and dynamic systems.

The implementation of the PointNet network in underwater scenarios presented some challenges, like ensuring its recognition performance when trained with point clouds obtained from underwater images, and its suitability to be integrated on an AUV due to its computational cost. With the results obtained in this work, we have demonstrated the validity of the PointNet deep neural network to detect pipes and valves in underwater scenarios for AUV manipulation and inspection tasks.

The datasets and code, along with one of the *Base* experiment trained models, are publicly available at UIB-SRV-3D-pipes for the scientific community to test or replicate our experiments.

Further steps need to be taken in order to achieve an underwater object localisation and positioning for ROV and AUV intervention using the object recognition presented in this work. We propose the following future work:

1. Performing an instance-based detection from the presented pixel-based one, allowing for recognition of pipes and valves as a whole object and to classify them by type (two or three way) or status (opened or closed).

2. Using the depth information provided by the stereo cameras along with the instance detection to achieve a spatial 3D positioning of each object. Once the network is implemented in an AUV, this would provide the vehicle with the information to manipulate and intervene with the recognised objects.

Acknowledgments

Miguel Martin-Abadal was supported by the Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contract DPI2017-86372-C3-3-R. Gabriel Oliver-Codina was supported by Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contract DPI2017-86372-C3-3-R. Yolanda Gonzalez-Cid was supported by the Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contracts TIN2017-85572-P and DPI2017-86372-C3-3-R; and by the Comunitat Autonoma de les Illes Balears through the Direcció General de Política Universitaria i Recerca with funds from the Tourist Stay Tax Law (PRD2018/34).

CRediT authorship contribution statement

Miguel Martin-Abadal: Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. **Manuel Piñar-Molina:** Software, Investigation, Data Curation, Writing - Original Draft. **Antoni Martorell-Torres:** Software, Data Curation. **Gabriel Oliver-Codina:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Yolanda Gonzalez-Cid:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

[1] Ahmed, S., Khan, M.F.R., Labib, M.F.A., Chowdhury, A.E., 2020. An observation of vision based underwater object detection and tracking, in: 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), pp. 117–122. doi:doi: 10.1109/ICETCE48199.2020.9091752.

[2] Asakawa, K., Kojima, J., Kato, Y., Matsumoto, S., Kato, N., 2000. Autonomous underwater vehicle aqua explorer 2 for inspection of underwater cables, in: Proceedings of the 2000 International Symposium on Underwater Technology (Cat. No.00EX418), pp. 242–247. doi:doi: 10.1109/UT.2000.852551.

[3] Bazeille, S., Quidu, I., Jaulin, L., 2012. Color-based underwater object recognition using water light attenuation. Intelligent Service Robotics 5, 109–118. doi:doi: 10.1007/s11370-012-0105-3.

[4] Burguera, A., Bonin-Font, F., 2020. On-line multi-class segmentation of side-scan sonar imagery using an autonomous underwater vehicle. Journal of Marine Science and Engineering 8, 557. doi:doi: 10.3390/jmse8080557.

[5] Camera Calibration Repository, . Ros - camera calibration. http://wiki.ros.org/camera_calibration. Accessed: 2020-12-07.

[6] Camera Info Repository, . Ros - camera info. http://wiki.ros.org/image_pipeline/CameraInfo. Accessed: 2020-12-07.

[7] Capocci, R., Dooly, G., Omerdić, E., Coleman, J., Newe, T., Toal, D., 2017. Inspection-class remotely operated vehicles—a review. Journal of Marine Science and Engineering 5, 13. doi:doi: 10.3390/jmse5010013.

[8] Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85. doi:doi: 10.1109/CVPR.2017.16.

[9] Chen, Z., Wang, H., Xu, L., Shen, J., 2014. Visual-adaptation-mechanism based underwater object extraction. Optics and Laser Technology 56, 119–130. doi:doi: 10.1016/j.optlastec.2013.07.003.

[10] Cheng, J.C., Wang, M., 2018. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. Automation in Construction 95, 155–171. doi:doi: 10.1016/j.autcon.2018.08.006.

[11] Costa, M., Pinto, J., Ribeiro, M., Lima, K., Monteiro, A., Kowalczyk, P., Sousa, J., 2019. Underwater archaeology with light auvs, in: OCEANS 2019 - Marseille, pp. 1–6. doi:doi: 10.1109/OCEANSE.2019.8867503.

[12] Denos, K., Ravaut, M., Fagette, A., Lim, H., 2017. Deep learning applied to underwater mine warfare, in: OCEANS 2017 - Aberdeen, pp. 1–7. doi:doi: 10.1109/OCEANSE.2017.8084910.

[13] Fatan, M., Daliri, M.R., Mohammad Shahri, A., 2016. Underwater cable detection in the images using edge classification based on texture information. Measurement: Journal of the International Measurement Confederation 91, 309–317. doi:doi: 10.1016/j.measurement.2016.05.030.

[14] FLIR, T., . Bumblebee 2 stereo rig specifications. https://www.flir.com/support/products/bumblebee2-firewire/#Overview. Accessed: 2022-10-22.

[15] Foresti, G.L., Gentili, S., 2002. A hierarchical classification system for object recognition in underwater environments. IEEE Journal of Oceanic Engineering 27, 66–78. doi:doi: 10.1109/48.989889.

[16] Geisser, S., 1975. The predictive sample reuse method with applications. Journal of the American Statistical Association 70, 320–328. doi:doi: 10.1080/01621459.1975.10479865.

[17] Guerra, E., Palacin, J., Wang, Z., Grau, A., 2020. Deep learning-based detection of pipes in industrial environments, in: Industrial Robotics - New Paradigms. IntechOpen. doi:doi: 10.5772/intechopen.93164.

[18] Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision. 2 ed., Cambridge University Press, USA.

[19] Heshmati-Alamdari, S., Bechlioulis, C.P., Karras, G.C., Nikou, A., Dimarogonas, D.V., Kyriakopoulos, K.J., 2018. A robust interaction control approach for underwater vehicle manipulator systems. Annual Reviews in Control 46, 315 – 325. doi:doi: https://doi.org/10.1016/j.arcontrol.2018.10.003.

[20] Heshmati-Alamdari, S., Nikou, A., Dimarogonas, D.V., 2021. Robust trajectory tracking control for underactuated autonomous underwater vehicles in uncertain environments. IEEE Transactions on Automation Science and Engineering 18, 1288–1301. doi:doi: 10.1109/TASE.2020.3001183.

[21] Himri, K., Pi, R., Ridao, P., Gracias, N., Palomer, A., Palomeras, N., 2018. Object recognition and pose estimation using laser scans for advanced underwater manipulation, in: 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), pp. 1–6. doi:doi: 10.1109/AUV.2018.8729742.

[22] Himri, K., Ridao, P., Gracias, N., 2019. 3d object recognition based on point clouds in underwater environment with global descriptors: A survey. Sensors 19, 4451. URL: http://dx.doi.org/10.3390/s19204451, doi:doi: 10.3390/s19204451.

[23] Inzartsev, A., Eliseenko, G., Panin, M., Pavin, A., Bobkov, V., Morozov, M., 2019. Underwater pipeline inspection method for AUV based on laser line recognition: Simulation results. 2019 IEEE International Underwater Technology Symposium, UT 2019 - Proceedings , 1–8doi:doi: 10.1109/UT.2019.8734387.

[24] Jacobi, M., Karimanzira, D., 2013. Underwater pipeline and cable inspection using autonomous underwater vehicles, in: 2013 MTS/IEEE OCEANS - Bergen, pp. 1–6. doi:doi: 10.1109/OCEANS-Bergen.2013.6608089.

[25] Jalal, A., Salman, A., Mian, A., Shortis, M., Shafait, F., 2020. Fish detection and species classification in underwater environments using deep learning with temporal information. Ecological Informatics 57, 101088. doi:doi: 10.1016/j.ecoinf.2020.101088.

[26] Jeon, M., Lee, Y., Shin, Y.S., Jang, H., Kim, A., 2019. Underwater Object Detection and Pose Estimation using Deep Learning. IFAC-PapersOnLine 52, 78–81. doi:doi: 10.1016/j.ifacol.2019.12.286.

[27] Jonsson, P., Sillitoe, I., Dushaw, B., Nystuen, J., Heltne, J., 2009. Observing using sound and light – a short review of underwater acoustic and video-based methods. Ocean Science Discussions 6, 819–870. doi:doi: 10.5194/osd-6-819-2009.

[28] Kallasi, F., Oleari, F., Bottioni, M., Lodi Rizzini, D., Caselli, S., 2014. Object detection and pose estimation algorithms for underwater manipulation, in: 2014 Conference on Advances in Marine Robotics Applications.

[29] Kim, B., Yu, S., 2017. Imaging sonar based real-time underwater object detection utilizing adaboost method, in: 2017 IEEE Underwater Technology (UT), pp. 1–5. doi:doi: 10.1109/UT.2017.7890300.

[30] Kim, D., Lee, D., Myung, H., Choi, H., 2012. Object detection and tracking for autonomous underwater robots using weighted template matching, in: 2012 Oceans - Yeosu, pp. 1–5. doi:doi: 10.1109/OCEANS-Yeosu.2012.6263501.

[31] Kumar, S.S., Abraham, D.M., Jahanshahi, M.R., Iseley, T., Starr, J., 2018. Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks. Automation in Construction 91, 273–283. doi:doi: 10.1016/j.autcon.

2018.03.028.

[32] Lee, D., Kim, G., Kim, D., Myung, H., Choi, H.T., 2012. Vision-based object detection and tracking for autonomous navigation of underwater robots. Ocean Engineering 48, 59–68. doi:doi: 10.1016/j.oceaneng.2012.04.006.

[33] Lee, S., Park, B., Kim, A., 2018. Deep learning from shallow dives: Sonar image generation and training for underwater object detection. ArXiv arXiv:1810.07990.

[34] Lee, S., Park, B., Kim, A., 2019. A deep learning based submerged body classification using underwater imaging sonar, in: 2019 16th International Conference on Ubiquitous Robots (UR), pp. 106–112. doi:doi: 10.1109/URAI.2019.8768581.

[35] Lodi Rizzini, D., Kallasi, F., Aleotti, J., Oleari, F., Caselli, S., 2017. Integration of a stereo vision system into an autonomous underwater vehicle for pipe manipulation tasks. Computers and Electrical Engineering 58, 560–571. doi:doi: 10.1016/j.compeleceng.2016.08.023.

[36] Lodi Rizzini, D., Kallasi, F., Oleari, F., Caselli, S., 2015. Investigation of vision-based underwater object detection with multiple datasets. International Journal of Advanced Robotic Systems 12, 1–13. doi:doi: 10.5772/60526.

[37] Martin-Abadal, M., Ruiz-Frau, A., Hinz, H., Gonzalez-Cid, Y., 2020. Jellytoring: Real-time jellyfish monitoring based on deep learning object detection. Sensors 20, 1–21. doi:doi: 10.3390/s20061708.

[38] Narimani, M., Nazem, S., Loueipour, M., 2009. Robotics vision-based system for an underwater pipeline and cable tracker, in: OCEANS 2009-EUROPE, pp. 1–6. doi:doi: 10.1109/OCEANSE.2009.5278327.

[39] Nikou, A., Verginis, C.K., Dimarogonas, D.V., 2018. A tube-based mpc scheme for interaction control of underwater vehicle manipulator systems, in: 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), pp. 1–6. doi:doi: 10.1109/AUV.2018.8729801.

[40] Nunes, A., Gaspar, A.R., Matos, A., 2019. Critical object recognition in underwater environment, in: OCEANS 2019 - Marseille, pp. 1–6. doi:doi: 10.1109/OCEANSE.2019.8867360.

[41] Olmos, A., Trucco, E., 2002. Detecting man-made objects in unconstrained subsea videos, in: British Machine Vision Conference, pp. 50.1–50.10. doi:doi: 10.5244/C.16.50.

[42] Ortiz, A., Simó, M., Oliver, G., 2002. A vision system for an underwater cable tracker. Machine Vision and Applications 13, 129–140. doi:doi: 10.1007/s001380100065.

[43] Palomer, A., Ridao, P., Youakim, D., Ribas, D., Forest, J., Petillot, Y., 2018. 3D laser scanner for underwater manipulation. Sensors 18, 1–18. doi:doi: 10.3390/s18041086.

[44] Prats, M., García, J.C., Wirth, S., Ribas, D., Sanz, P.J., Ridao, P., Gracias, N., Oliver, G., 2012. Multipurpose autonomous underwater intervention: A systems integration perspective, in: 2012 20th Mediterranean Conference on Control Automation (MED), pp. 1379–1384. doi:doi: 10.1109/MED.2012.6265831.

[45] Prechelt, L., 2012. Early Stopping — But When?. Springer. pp. 53–67. doi:doi: 10.1007/978-3-642-35289-8_5.

[46] Rayhana, R., Jiao, Y., Liu, Z., Wu, A., Kong, X., 2020. Water pipe valve detection by using deep neural networks, in: Smart Structures and NDE for Industry 4.0, Smart Cities, and Energy Systems, SPIE. pp. 20 – 27. doi:doi: 10.1117/12.2558886.

[47] Rekik, F., Ayedi, W., Jallouli, M., 2018. A trainable system for underwater pipe detection. Pattern Recognition and Image Analysis 28, 525–536. doi:doi: 10.1134/S1054661818030185.

[48] Ridao, P., Carreras, M., Ribas, D., Sanz, P.J., Oliver, G., 2015. Intervention auvs: The next challenge. Annual Reviews in Control 40, 227–241. doi:doi: 10.1016/j.arcontrol.2015.09.015.

[49] Stereo Proc Repository, . Ros - stereo image proc. http://wiki.ros.org/stereo_image_proc. Accessed: 2022-05-18.

[50] Tascini, G., Zingaretti, P., Conte, G., 1996. Real-time inspection by submarine images. Journal of Electronic Imaging 5, 432– 442. doi:doi: 10.1117/12.245766.

[51] Wang, X., Liu, S., Liu, Z., 2017. Underwater sonar image detection: A combination of nonlocal spatial information and quantum-inspired shuoed frog leaping algorithm. PLoS ONE 12, 1–30. doi:doi: 10.1371/journal.pone.0177666.

[52] Yu, M., Ariamuthu Venkidasalapathy, J., Shen, Y., Quddus, N., Mannan, M.S., 2017. Bow-tie analysis of underwater robots in offshore oil and gas operations, in: Offshore Technology Conference. doi:doi: 10.4043/27818-MS.

[53] Zingaretti, P., Zanoli, S.M., 1998. Robust real-time detection of an underwater pipeline. Engineering Applications of Artificial Intelligence 11, 257–268. doi:doi: 10.1016/S0952-1976(97)00001-8.

Real-time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks

Miguel Martin-Abadal*, Gabriel Oliver-Codina and Yolanda Gonzalez-Cid

Department of Mathematics and Computer Science. University of the Balearic Islands, 07122, Palma, Spain

ARTICLE INFO

The work presented in this preprint has been published in the journal *Sensors* as:

Martin-Abadal, M.; Oliver-Codina, G.; Gonzalez-Cid, Y. *Real-Time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks*. Sensors 2022, 22, 8141.
DOI: 10.3390/s22218141

ABSTRACT

Nowadays, more frequently, it is necessary to perform underwater operations like surveying an area or inspecting and intervening on industrial infrastructures such as offshore oil and gas rigs or pipeline networks. Recently, the use of Autonomous Underwater Vehicles (*AUV*) has grown as a way to automate these tasks, reducing risks and execution time. One of the used sensing modalities is vision, providing RGB high-quality information in the mid to low range, making it appropriate for manipulation or detail inspection tasks. This work presents the usage of a deep neural network to perform pixel-wise 3D segmentation of pipes and valves on underwater point clouds generated using a stereo pair of cameras. In addition, two novel algorithms are built to extract information from the detected instances, providing pipe vectors, gripping points, the position of structural elements like elbows or connections, and valve type and orientation. The information extracted on spatially referenced point clouds can be unified to form an information map of an inspected area. Results show outstanding performance on the network segmentation task, achieving a mean *F1-score* value of 88.0% at a pixel-wise level and of 95.3% at an instance level. The information extraction algorithm also showcased excellent metrics when extracting information from pipe instances and their structural elements and good enough metrics when extracting data from valves. Finally, the neural network and information algorithms are implemented on an *AUV* and executed in real-time, validating that the output information stream frame rate of $0.72 fps$ is high enough to perform manipulation tasks and to ensure full seabed coverage during inspection tasks. The used dataset, along with a trained model and the information extraction and mapping algorithms, are provided to the scientific community in http://srv.uib.es/3d-pipes-2/.

1. Introduction

The need for conducting underwater intervention tasks has grown significantly in recent decades. More often it is necessary to perform underwater operations in different fields like archaeology, biology, rescue and recovery or industry that include not only inspection but also interaction with the environment. One of the most relevant cases concerns to manipulation tasks performed on offshore oil and gas rigs or pipeline networks [2, 5, 10, 22, 54].

In the past, the aforementioned tasks were mostly carried out, manually, by scuba divers. Nonetheless, conducting these missions in a hard-to-reach scenario like open waters tends to be slow, dangerous and resource consuming. Recently, Remotely Operated Vehicles (*ROVs*) equipped with diverse sensing systems and manipulators have been used to access deeper and more complex underwater scenarios, allowing to eliminate some of the drawbacks of human intervention.

However, *ROVs* still presented downsides such as its hard and error-prone piloting due to complex water dynamics, requiring trained operators; or the need of a support vessel, leading to expensive operational costs. To ease these drawbacks, there has been increasing research towards intervention Autonomous Underwater Vehicles (*AUVs*) [19, 32, 48] and Underwater Vehicle Manipulator Systems (*UVMS*) [18, 38].

Other challenges faced in underwater environments are presented regarding its sensing in general and object perception in particular. Underwater sensing presents several challenges like distortion in signals, light absorption and scattering, water turbidity changes or depth-depending colour distortion.

Intervention *ROVs* and *AUVs* are often equipped with a variety of sensing systems. When operating in unknown underwater environments, sonar systems are usually preferred as it is able to obtain bathymetric maps of large areas in a short time. Even though sonar is mostly used to provide general information of the environment or used in a first stage approach to the area of interest, it also has been used to perform object detection by itself. Nonetheless, the preferred sensing modalities to obtain detailed, short-distance information with higher resolution are laser and video. These modalities are often used during the approach, object recognition and intervention phases. The usage of the presented sensing systems towards object detection and, specifically, pipe and valve recognition is reviewed in Section 2.1.

To execute manipulation tasks, the sensing systems of a *ROV* or *AUV* must be able to provide enough information to perform accurate and robust scene understanding, including object detection, target recognition and pose estimation, under different experimental conditions.

*Corresponding author
miguel.martin@uib.es (M. Martin-Abadal)

This paper is a continuation of our previous work [35] where we proposed a deep learning-based approach to perform a pixel-wise segmentation of underwater pipes and valves from 3D RGB point cloud information.

In this paper, we make use of an improved evolution of the deep neural network used in our previous work to perform the pixel-wise 3D segmentation. Additionally, we implement an object detection over the segmented pixels, grouping them and being able to detect diverse pipe and valve instances in a point cloud. We develop an algorithm to extract information from the detected instances providing pipe vectors, gripping points, structural elements like elbows or connections, and valve type (2-way or 3-way) and orientation. Furthermore, if the point clouds are spatially referenced, its information can be unified, forming an information map of an inspected area. Finally, the 3D segmentation, along with the information extraction and mapping algorithms, are executed online on an *AUV*, performing real-time underwater pipe and valve recognition, characterisation and mapping for inspection and manipulation tasks.

The remainder of this paper is structured as follows: Section 2 reviews the related work on underwater perception and pipe and valve identification, and highlights the main contributions of this work. Section 3 describes the methodology and materials adopted in this study. The experimental results are presented and discussed in Section 4. Section 5 details the network and information algorithms online implementation. Finally, Section 6 outlines the main conclusions and future work.

2. Related Work and Contributions

2.1. State of the Art

This section reviews the usage of diverse sensing systems for underwater inspection, object detection and, specifically, pipe and valve recognition. The three most used sensing systems in underwater environments are sonar, laser and vision.

Sonar sensing is the preferred method when working in large, unknown environments, providing broad information in a quick manner [4, 25]. It has also been used for object localisation in underwater scenarios [27, 52]. Object detection deep learning techniques have been applied to underwater sonar imaging for diverse applications like the detection of human bodies [31] or war mines [11]. Some of the drawbacks presented by the sonar imaging are the noisy nature of the images, which generates texture information losses; and the fact that it is not able to capture colour information, which is useful in object recognition tasks.

Underwater laser scans can provide high resolution 3D data that can be used for environment inspection and object recognition. Some studies on underwater pipeline detection include the works of Palomer et al. [42] where a laser scanner is integrated on an AUV for object detection and manipulation, or the works of Himri et al. [20, 21] and Villacrosa et. al. [51], where a recognition and pose estimation pipeline based on point cloud matching is built. As for downsides, laser systems tend to have a very high initial cost, are affected by light transmission problems and neither can provide colour information.

Vision is one of the most complete and used perception modalities in robotics and object recognition tasks thanks to its accessibility, easiness to use and the fact that produces RGB high-quality information. It also has disadvantages as the obtained images are affected by light transmission problems, colouring distortions or environmental factors such as water turbidity. Nonetheless, some of these weaknesses can be alleviated by adapting the acquisition system to the environmental conditions, adjusting the operation range, calibrating the cameras or colour correcting the obtained images.

In the past, traditional computer vision approaches have been used to detect and track multiple submerged objects such as artifacts [1, 8, 40, 43], cables [13, 37, 41] or pipelines [15, 37, 50, 55]. Some works rely on texture and shape descriptors [15, 37], others on template matching [28, 30] or use colour segmentation to find and process regions of interest in the images [3, 43].

Other works use a combination of multiple sources of information, Kallasi et al. in [26] and Razzini et al. in [32, 33] present traditional computer vision methods that combine texture, shape and colour information to detect underwater pipelines and project them into point clouds obtained from stereo vision. In these works, the point cloud information is not used to assist the pipe recognition process.

Rekik et al. [47] developed the first trainable system to detect underwater pipelines, extracting several features and using a Support Vector Machine to classify between positive and negative underwater pipe images samples. Convolutional Neural Networks were introduced by Nunes et. al [39] to classify diverse underwater objects, including a pipeline. None of these works determined the position of the object within the image, only a binary classification of the object's presence was given.

Some studies introduced deep learning solutions applied to underwater computer vision, but are limited to the detection and pose estimation of 3D-printed objects [24] or living organisms like fishes [23] or jellyfishes [36]. Few research studies involving pipelines are restricted to damage evaluation [9, 29] or pipeline navigation from the inside [46]. Guerra et al. in [17] presents one of the most advanced works on pipeline recognition using deep learning, where a drone equipped with a monocular camera is used to perform 2D detection of pipelines in industrial environments.

Therefore, with the exception of the later works of Himri et al. [21] and Villacrosa et al. [51], which will be later discussed in Section 4.1, the remaining works suffer from crucial drawbacks when tackling pipe and valve recognition for inspection and manipulation tasks. The most significant drawbacks from previous implementations, which are solved in our work, are listed below:

– Only recognising pipes, no valves, connections or elbows are detected.

– Not being able to detect multiple elements simultaneously, due to the nature of its data processing, only isolated objects can be detected.

– Not gathering information from the detected objects, such as pipe length, gripping points, orientation or valve type and position.

– Not being able, or no demonstration, of being able to be executed online on a inspecting or manipulating robot.

Finally, to the best knowledge of the authors, the only prior know research on underwater pipeline and valve 3D recognition using deep learning is our previous work presented in [35], upon which we build the research introduced in this paper.

2.2. Main Contributions

The main contributions of this paper are composed of:

1. Expansion of our novel point cloud dataset of underwater pipe and valve structures, adding point clouds obtained with a new pair of cameras mounted on an *AUV*. This dataset is used to train and test the selected deep neural network and information algorithms.
2. Development of novel algorithms to extract information from detected pipe and valve instances, providing data on pipe vectors, gripping points, structural elements like elbows or connections, and valve type and orientation. The information from spatially referenced point clouds can be unified to create information maps of inspected areas.
3. Neural network and information algorithms validation by conducting underwater experiments where the point cloud segmentation, the information extraction algorithm and the mapping algorithm are executed online in an *AUV*, performing real-time underwater pipe and valve recognition, characterisation and mapping for inspection and manipulation tasks.
4. The updated dataset (point clouds and corresponding ground truths) along with a trained model and the code of the algorithms used to perform the information extraction and mapping are provided to the scientific community in [34].

3. Methodology

This section presents an overview of the selected network; explains the acquisition, labelling and organisation of the data; details the information extraction and mapping algorithms; and exposes the validation process and evaluation metrics.

3.1. Deep Learning Network and Training Deatils

Even though most applications in the field work with 2D information, which some later project to the 3D space, we decided to use a 3D segmentation network using point clouds as input for diverse reasons. First of all, the introduction of depth data provides extra information to work with, allowing to extract more features, helping to the segmentation. Secondly, as we extract information from the segmented point clouds for inspection and manipulation tasks, 3D positioning would be a must. Thus, it would not make sense to use 2D segmentation to avoid possible matching failures in the 3D point cloud generation if the extracted information could not be projected into a 3D space.

In this work, we select the Dynamic Graph Convolutional Neural Network (*DGCNN*) [53] to perform the pipe and valve 3D segmentation. This network is an evolution of the PointNet deep neural network [7] that we used in our previous work, surpassing its performance on several benchmark datasets [53]. Like its predecessor, this network has a unified architecture that allows it to perform multiple tasks, ranging from object classification and part segmentation to scene semantic segmentation.

The novelty of the *DGCNN* architecture is the introduction of the named EdgeConv modules, which can be integrated into existing deep learning models such as PointNet. These modules capture local geometric structure information by generating edge features that describe relations between a point and its neighbours, while being invariant to input permutations. Since proximity in the feature space differs from proximity in the input point cloud, the set of neighbours of a point changes from layer to layer. This results in a network graph that is updated after each network layer, leading to non-local diffusion of information over the whole point cloud. This allows the EdgeConv modules to capture global shape information.

Furthermore, to make the prediction invariant to the point cloud geometric transformations, all input sets are aligned to a canonical space before feature extraction. To achieve this a 3x3 matrix is applied, obtained from a tensor concatenating the coordinates of each point and the coordinate difference between its neighbours.

The *DGCNN* architecture takes point clouds as input and outputs a class label for each point. While training the network, it is also fed with ground truth labels, indicating the real class of each point from the point cloud. The labelling process is further detailed in Section 3.2.2.

As the original *DGCNN* implementation, we use a softmax cross-entropy loss along with an Adam optimiser. The decay rate for batch normalisation starts with 0.5 and is gradually increased to 0.99. In addition, we apply a dropout with a keep ratio of 0.7 on the last fully connected layer, before class score prediction.

Other hyperparameters are selected based on the experiments conducted on our previous work [35]. For the training, we use stochastic gradient descent with a learning rate of 0.001, a batch size of 16; block and stride distances of 1 meter; and, finally, a maximum number of allowed points per block of 128. These hyperparameters have proven to offer very good metrics in terms of point cloud segmentation while greatly reducing inference time, a key factor in the online execution of this network in an *AUV* to perform real-time inspection and manipulation tasks. Details on the

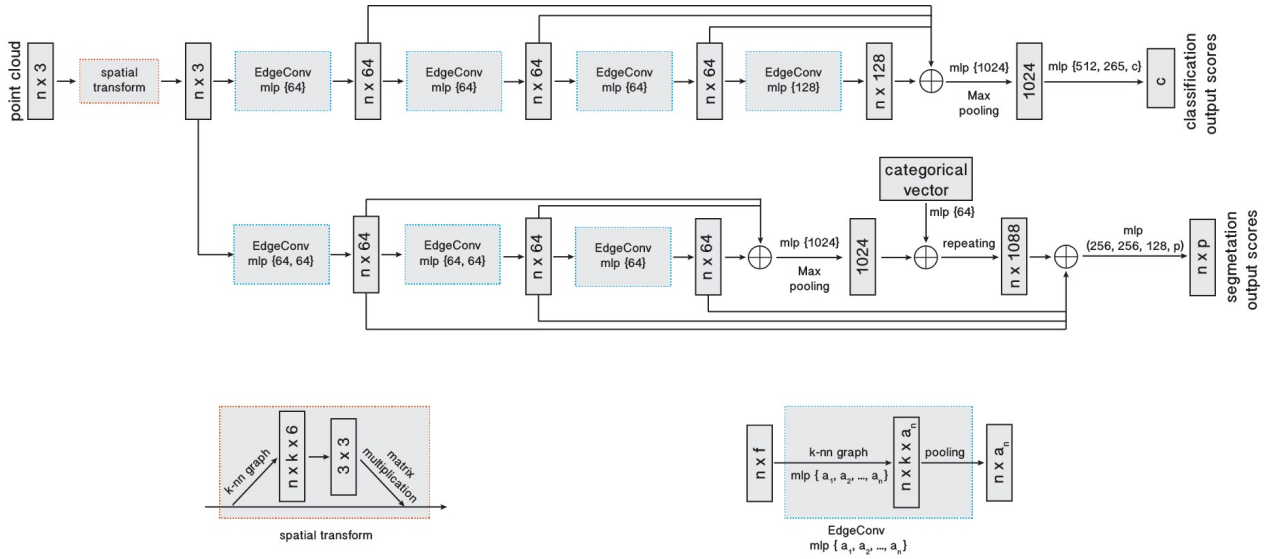


Figure 1: *DGCNN* architecture. Taken from [53], with permission from author Yue Wang, 2021.

network and algorithms online execution in an *AUV* are given in Section 5.

To improve the network performance, we implement an early stopping strategy based on the work of L. Prechelt in [44], ensuring that the network training process stops when the divergence between validation and training losses is minimum. This technique allows to obtain a more general and broad training, avoiding overfitting. The *DGCNN* architecture is presented in Figure 1.

3.2. Data

This subsection explains the acquisition, labelling and managing of the data used to train and test the deep neural network.

3.2.1. Acquisition

The used point clouds come from two different sources. First, we reuse our previous dataset from [35], consisting of 192 underwater point clouds containing diverse pipe and valve structures and connections. This dataset was gathered on an artificial pool using a Bumblebee2 Firewire stereo rig mounted on an Autonomous Surface Vehicle and using the Robot Operating System (*ROS*) middleware [45].

The second source of point clouds is another stereo pair rig, composed of two Manta G283 cameras mounted on an *AUV*, gathering point clouds through *ROS* once again. We performed up to six immersions with the AUV to record different pipe structures and valve connections, two of them at an artificial pool, and the remaining four at different locations at the sea. The acquisition process is pictured in Figure 2.

3.2.2. Ground Truth Labelling

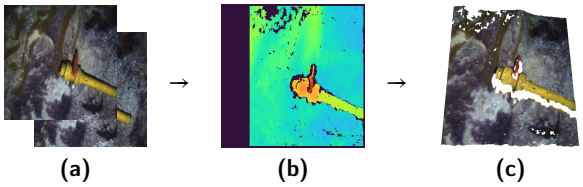
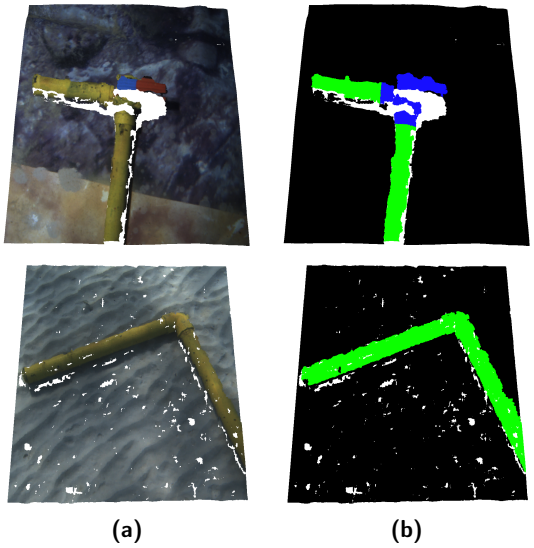


Figure 2: Data acquisition process. **(a)** left and right stereo images from a calibrated stereo rig. **(b)** Disparity depth image obtained using *ROS* stereo processing [49]. **(c)** Point cloud generated by merging depth and colour information (Blank spaces correspond to areas where no matching between stereo images could be found or to covered areas).

In order to train and test the network, ground truth label maps are manually built from the obtained point clouds. The points corresponding to each class are marked with a different label. The studied classes and their RGB labels are: *Pipe* (Green: 0, 255, 0), *Valve* (Blue: 0, 0, 255) and *Background* (Black: 0, 0, 0). Figure 3 shows a couple of point clouds along with their corresponding ground truth label maps.

3.2.3. Dataset Managing

To configure our dataset, we gather the point clouds obtained from the two previously mentioned sources in Section 3.2.1. First, we take 192 point clouds from our previous dataset (from now on, referred as set S_{ASV}). And second, we extract point clouds from the *AUV* immersions. From the two pool immersions we extract 104 and 51 point clouds (from now on, referred as sets $S_{POOL\text{-}1/2}$, respectively). From the four sea immersions, we extract 45, 56, 36 and 30 point clouds (from now on referred as sets $S_{SEA\text{-}1/2/3/4}$, respectively).



(a) (b)

Figure 3: Point cloud labelling. (**a**) Original point cloud. (**b**) Ground truth annotations. Points corresponding to pipes, valves and background, are marked in green, blue and black, respectively.

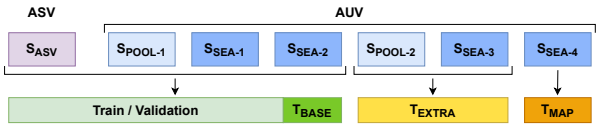


Figure 4: Dataset managing.

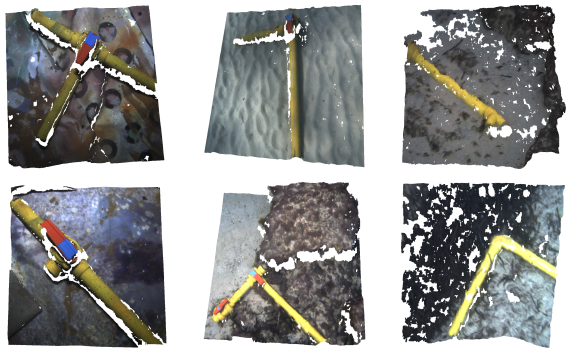


Figure 5: Point clouds from the dataset showcasing diverse pipe structures and valve connections over different backgrounds.

In order to build the dataset used for the training, validation and test, we decide to gather the point clouds from the sets S_{ASV}, $S_{POOL\text{-}1}$ and $S_{SEA\text{-}1/2}$, conforming a total of 397 point clouds along their corresponding ground truth label maps.

This dataset contains point clouds gathered using two different pairs of stereo cameras, in fresh and salt water, under different environmental conditions and showcasing a wide variety of pipe structures and valve connections over different backgrounds, such as a plastic lone, sand or rocks. This represents a broad spectrum of scenarios to assure robustness in the network training and reduce its overfitting. The dataset is split into a train-validation set (90% of the data, 357 point clouds) and a test set (10% of the data, 40 point clouds), which will be referred as T_{BASE}. Additionally, sets $S_{POOL\text{-}2}$ and $S_{SEA\text{-}3}$ are used to perform a secondary test, from now on referred as T_{EXTRA}, containing a total of 87 point clouds. The point clouds from this test are from immersions whose data has not been used for the network training or validation, and contain different, unseen environmental conditions, pipe structures, valve connections and backgrounds. Hence, this test allows to assess how well the network generalises its training to new data.

Finally, set $S_{SEA\text{-}4}$ contains point clouds gathered by the *AUV* navigating over a larger structure containing multiple pipes and valves. This set will be used to test the mapping algorithm presented in Section 3.3. This test set will be referred as T_{MAP}.

Figure 4 illustrates the dataset managing, while in Figure 5 some examples of point clouds are shown.

3.3. Segmentation Understanding Algorithms

Once the deep neural network has processed a point cloud and generated its semantic segmentation, we need to further process this segmentation and extract information to use it in inspection and manipulation tasks. To do that, we develop two algorithms. The first algorithm, referred as Information Extraction Algorithm (*IEA*), takes the network output and extracts information such as the number of pipes and valves present in the point cloud, its position, orientation or even pipe connections and valve type (2-way or 3-way). The second algorithm, referred as Information Unification Algorithm (*IUA*), unifies the information extracted from multiple localised point clouds taken on a studied area and generates a global information map. Next, both *IEA* and *IUA* algorithms are detailed.

3.3.1. Information Extraction Algorithm

The starting point of this algorithm is the semantic segmentation outputted by the deep neural network. Its first step is to transform the pixel-wise segmentation into an instance-based one using a Density-Based Spatial Clustering of Applications with Noise (*DBSCAN*), clustering pixels of the same class that are closer than a distance threshold, clusters that do not contain enough points to be considered as instances are deleted. This way, the different pipe and valve instances present in a point cloud are detected. Additionally, when a cluster belonging to a valve instance is found, it is set to "steal" the points belonging to pipe instances that are within a determined radius of the valve instance central point. This is due to the fact that, as the main body of the valve is very similar to the actual pipes, it sometimes gets missclassified as pipe and only the handle of the valve is correctly classified, this way, the body of the valve is reclassified. Figure 6 shows the point clustering and valve reclassification on a segmented point cloud.

The following step of the algorithm is to extract information from the detected valve instances. First, the central

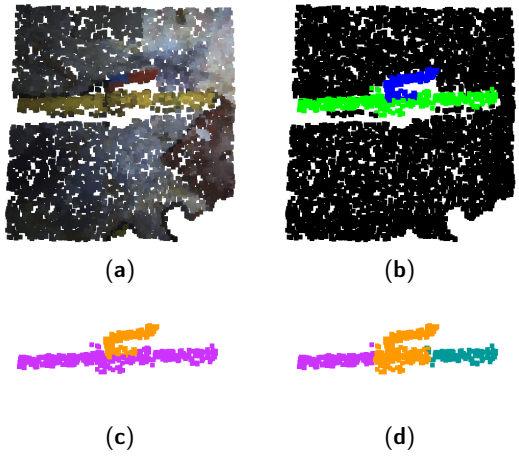


Figure 6: Point cloud clustering and valve reclassification. **(a)** Original point cloud. **(b)** Deep neural network segmentation. **(c)** Pipe and valve point clustering. **(d)** Valve reclassification.

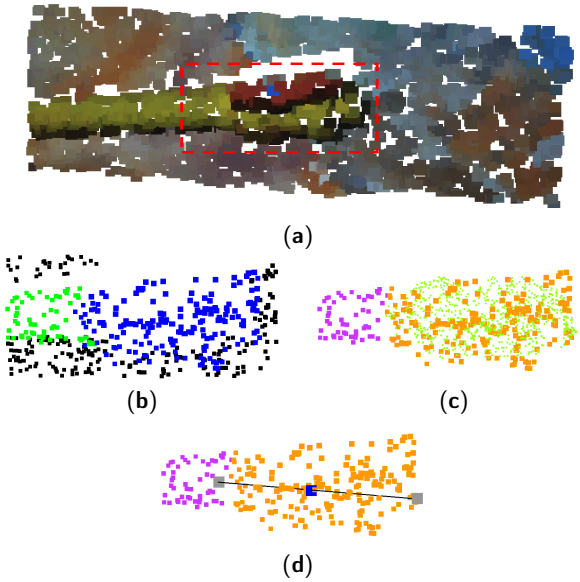


Figure 7: Valve information extraction process. **(a)** Original point cloud (area of interest highlighted in red square). **(b)** Deep neural network segmentation. **(c)** Instance clustering and best valve model registration (light green points). **(d)** Instance clustering along resulting valve central point (blue point) and vector (black line between the two gray points).

point of the valve instances is calculated as the average XYZ coordinate values of all its belonging points, noting the valve position. Second, the algorithm performs a point cloud registration between each valve instance and five point cloud models of 2-way and 3-way valves, obtaining the rotation matrix and model that provides a maximum registration score, inferring its pose and type. Using the valves registration data and its pre-known shape, a vector is generated to indicate each valve size and orientation.

Valve instances that do not reach a certain registration score threshold with any point cloud model are discarded. Consequently, the previously mentioned "stolen" points corresponding to discarded valve instances are returned to their corresponding original pipe instances.

Figure 7 showcases the described valve information extraction process.

The next step is to extract information from the detected pipe instances. First, the point cloud instances are voxelized and flattened into a 2D matrix, where closing and opening morphology operations are performed to consolidate the instance as a unique object.

At this point, the skeleton of the instance is computed, obtaining a chain of linked matrix coordinates, depicting the pipe shape. Also, coordinates with up to three neighbours are marked as connection points between different pipes. Once the smaller chains are discarded, the remaining chains and connection points are reprojected into the original instance 3D points.

Then, the algorithm calculates the 3D vector between each chain point and its linked points, providing information about the chain curvature. From there, pipe elbows can be established on point sequences with greater curvature than a selected threshold. Also, these vectors provide information on the chain length, allowing to locate the position of a determined percentage of pipe length, this information is very useful to provide grabbing points for pipe manipulation. Finally, a vector describing each straight portion of a chain is calculated between its first and last point, giving information on the corresponding pipe orientation and length.

Figure 8 showcases the described pipe information extraction process.

Finally, the valve and pipe information is refined. For the valves, its vector direction is recalculated taking into account the presence of pipes near the valve central point. If only one pipe is near, the valve vector is aligned to the pipe vector, if two or three pipes are present and two of them have parallel vectors, the valve pipe is aligned with that vector. Additionally, if three pipes are found near its central point, the valve type is set automatically to 3-way. For the pipes, vectors belonging to pipes of different instances that are near and parallel, are unified.

Figure 9 shows examples of valve and pipe refinement.

3.3.2. Information Unification Algorithm

This algorithm is built on top of the information provided by the previously detailed *IEA* and its end is to generate unified information maps by merging information from different point clouds. It is strictly necessary that the point clouds are referenced to a localised frame, whether it is an absolute frame like geolocalisation or a relative one such as odometry.

Different methods are used to merge the pipe and valve information from new upcoming point clouds to the one already present in the information map. For the pipes, the algorithm checks if upcoming pipe chains are near chains present in the information map. Near chains are merged and new vectors and elbows are computed as explained in

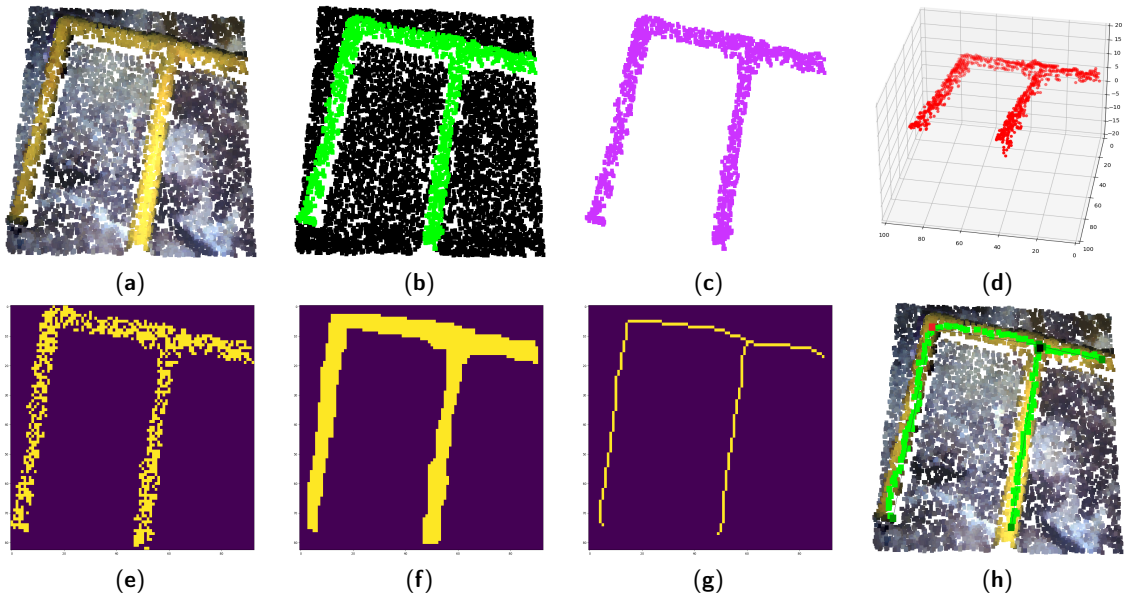


Figure 8: Pipe information extraction process. **(a)** Original point cloud. **(b)** Deep neural network segmentation. **(c)** Instance clustering. **(d)** 3D voxelization. **(e)** 2D matrix flattening. **(f)** Morphological operations: closing and opening. **(g)** Morphological operation: skeletonization. **(h)** Information reprojection overlapped onto original point cloud (light green: skeleton, dark green: pipe start-end, red: elbow, black: connection, lines: pipe vectors).

the *IEA* description. Also, a validation count is assigned to each pipe chain, indicating the number of times it has appeared in different point cloud information extractions, merged chains add up their validation counts. This validation count is used further into the algorithm to decide whether or not a detection is a spurious false positive or a certain true positive.

For the connection points, the algorithm checks if new upcoming connections are situated near prior registered connections. Near connections are merged, averaging its 3D position. A validation count is also assigned to each connection point.

For the valves, the procedure is the same as for the connection points, with the addition that the valve vector direction is also averaged between the prior valve vector and the upcoming one.

Finally, each K processed point clouds, the algorithm runs a validation count check, where pipes, connection points and valves with lower validation count than a determined threshold are discarded.

Figure 10 presents the *IUA* output when implemented over a series of point clouds containing two pipes, a valve and an elbow. It can be seen how a false positive valve appears on the original information extracted near the elbow, but it is later eliminated by the algorithm count check as it is not found in any other point cloud information.

Figure 11 presents a flowchart of both *IEA* and *IUA* algorithms, describing their workflow and interrelation. Additionaly, the commented code of both algorithm implementations is provided in [34], which offers a deeper insight into the diverse algorithms steps and their numerous parameters that can be tweaked.

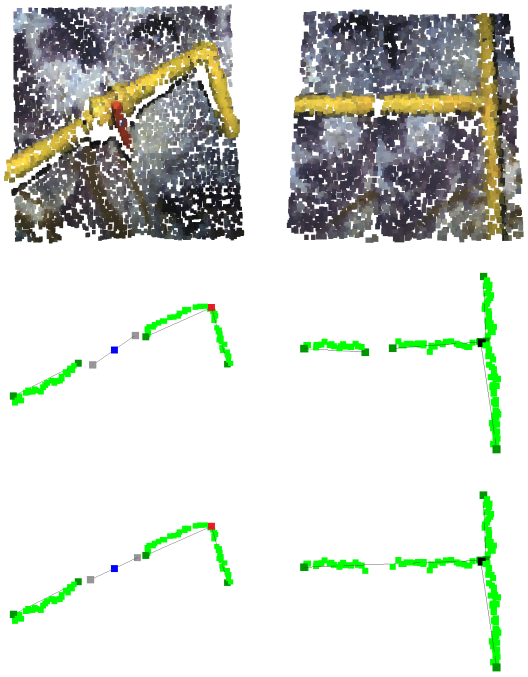
3.4. Validation and Evaluation Metrics

DGCNN is a highly efficient and effective network, obtaining great metrics in both object classification and segmentation tasks in indoor and outdoor scenarios [53]. However, it has never been tested and validated in underwater scenarios.

In order to validate the *DGCNN*, we use the 10k-fold cross-validation method [16]. Whit it, the train-validation partition of our dataset (see Figure 4) is split into ten equally sized subsets. Next, the network is trained ten times, each one using a different subset as validation and the nine remaining as training, generating ten models, which are then tested against the T_{BASE} and T_{EXTRA} test sets. The final performance is computed as the average results for the ten models. This method reduces the variability of the results, making them less dependent on the selected training and validation data, and therefore obtaining a more accurate performance estimation.

To evaluate a model performance we make a point-wise comparison between the network predictions and their corresponding ground truth annotations. For each class, the number of correctly segmented points, True Positives (*TP*); and the number of incorrectly segmented points, False Positives (*FP*) or False Negatives (*FN*) is computed. The number of *TP*, *FP* and *FN* are used to calculate the *Precision*, *Recall* and *F1-score* for each class, following Equations (1), (2) and (3).

$$Precision = \frac{TP}{TP + FP} \quad , \tag{1}$$



(a) (b)

Figure 9: Information refinement. **(a)** Valve vector reorientation, **(b)** Pipe vector unification. Top: Original point cloud. Middle: information before refinement. Bottom: information after refinement.

$$Recall = \frac{TP}{TP + FN} \quad , \tag{2}$$

$$F1\text{-}score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \quad . \tag{3}$$

The goal of this work is not to work on a pixel-level segmentation, but to group pixels into instances from which extract information. Therefore, it makes sense to evaluate the network performance on an instance level as well. In order to achieve that, we apply the clustering method explained in Section 3.3.2 to the network segmentation output and ground truth annotations. From there, we make use of the *Intersection over Union* (IoU) metric, which provides the similarity between two instances. The IoU value between two instances is calculated following Equation (4).

$$IoU = \frac{inst1 \cap inst2}{inst1 \cup inst2} = \frac{P_{shared}}{P_{inst1} + P_{inst2} - P_{shared}} \quad . \tag{4}$$

Where $P_{inst1/2}$ denotes the number of points forming instance one or two and P_{shared} the number of points shared by both instances.

To determine whether a predicted instance is a *TP* or a *FP*, an IoU threshold value needs to be established. Following the criteria applied in the PASCAL VOC challenge [12],

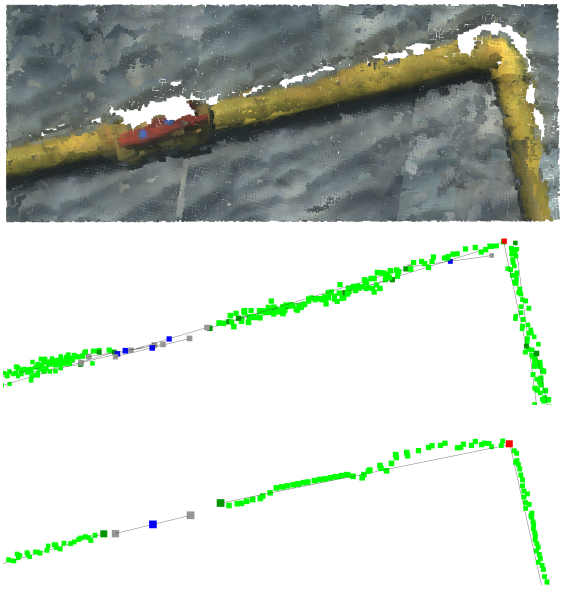


Figure 10: *IUA* implementation. Top: Overlapped point clouds. Middle: Overlapped extracted information from *IEA*. Bottom: Unified information from *IUA*.

we set this threshold at $thr_{iou} = 0.5$. A predicted instance is classified as *TP* if the IoU value with any ground truth instance is greater than the thr_{iou} and the prediction class (C_{pred}) is the same as the ground truth instance class (C_{gt}). Otherwise, the predicted instance is classified as a *FP* (Equation (5)).

$$Inst. = \begin{cases} TP, & \text{if } IoU >= thr_{iou} \ \& \ C_{pred} == C_{gt} \\ FP, & \text{otherwise} \end{cases} , \tag{5}$$

Ground truth instances that do not have an IoU>thr_{iou} with any predicted instance are counted as undetected instances, *FN*.

Once each prediction instance is classified as either *TP* or *FP*, and the number of *FN* is obtained, the instance-level *Precision*, *Recall* and *F1-score* metrics are computed following the previous Equations (1), (2) and (3).

Finally, to evaluate the information provided by the *IEA* and *IUA*, information ground truths are built, manually, over the T_{BASE}, T_{EXTRA} and T_{MAP} test sets point cloud network segmentations, annotating the same information generated by the *IEA*. The extracted information is compared to the ground truth annotations using different metrics for each type of information.

For the pipe information, the vectors are compared in terms of magnitude and direction, for the elbows and connections points the distance difference from their ground truth annotation counterparts is measured.

For the valve information, the describing vector is compared only in terms of direction, since its magnitude is fixed by parameter, central valve point diversion is also measured. Lastly, the correct valve type classification is checked.

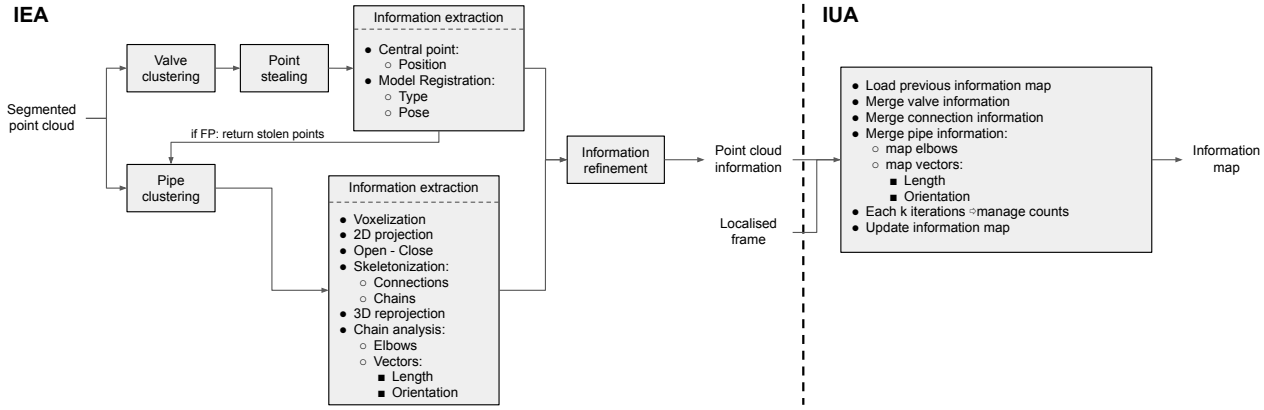


Figure 11: *IEA* and *IUA* algorithms flowchart, describing their workflow and interrelation.

Table 1
DGCNN pixel-level metrics on T_{BASE} test set.

Class	*F1(%)*	*mF1(%)*	*2mF1(%)*
Pipe	92.4		
Valve	84.9	92.3	**88.7**
Background	99.7		

Table 2
DGCNN instance-level metrics on T_{BASE} test set.

Class	*IoU(%)*	*mIoU(%)*	*F1(%)*	*mF1(%)*
Pipe	83.4	80.4	96.7	**95.1**
Valve	77.4		93.5	

Table 3
DGCNN pixel-level metrics on T_{EXTRA} test set.

Class	*F1(%)*	*mF1(%)*	*2mF1(%)*
Pipe	91.4		
Valve	83.0	91.4	**87,2**
Background	99.7		

Table 4
DGCNN instance-level metrics on T_{EXTRA} test set.

Class	*IoU(%)*	*mIoU(%)*	*F1(%)*	*mF1(%)*
Pipe	82.7	79.7	96.3	**95.4**
Valve	76.8		94.6	

4. Experimental results and Discussion

This section reports the results obtained by the *DGCNN* segmentation at pixel and instance levels. The *IEA* and *IUA* are also evaluated, checking the validity of the proportioned information.

4.1. *DGCNN* Segmentation Results

The results presented in this section are the average values obtained from the ten models generated when using the 10k-fold cross-validation method previously explained in Section 3.4.

Table 1 presents the pixel-level metrics when evaluating the *DGCNN* over the T_{BASE} test set. The presented metrics are the per-class *F1-score (F1)* and its mean value *(mF1)*. Additionally, since the background class is not converted into instances nor used by the *IEA* or *IUA*, it makes sense to only focus on the pipe and valve classes when conducting the per-pixel evaluation of the network. The *2mF1* metric measures the mean *F1-score* only taking into account those two classes.

The *DGCNN* reaches a *F1-score* value of 92.4% for the pipe class, an 84.9% for the less represented and harder to identify valve class and a 99.7% for the prevailing background class, resulting in a mean *F1-score* of 92.3%. When only taking into account the pipe and valve classes, the network scores an *2mF1* of 88.7%.

Table 2 presents the instance-level metrics when evaluating the *DGCNN* over the T_{BASE} test set. The presented metrics are the per-class Intersection over Union (*IoU*) and its mean value (*mIoU*) along the per-class *F1-score (F1)* and its mean value *(mF1)*.

The achieved *mIoU* value is 80.4%, which indicates a high overlap between predicted and ground truth instances. This similarity is reflected on the *mF1* score, reaching a value of 95.1%. The reached instance-level *mF1* score is higher than the obtained pixel-level *2mF1* score, this indicates that the applied pixel clustering allows to match predicted and ground truth instances even when there exist pixel differences between them.

Tables 3 and 4 present the pixel-level and instance-level metrics, respectively, when evaluating the *DGCNN* over the T_{EXTRA} test set.

The results for the T_{EXTRA} test set at both pixel-level and instance-level evaluations are equally good as the ones obtained for the T_{BASE} test set, reaching a pixel-level *2mF1* of 87.2% and an instance-level *mF1* of 95.4%. This means that the *DGCNN* is able to generalise its training and avoid overfitting, being able to correctly segment more challenging point clouds with unseen pipe and valve connections and

Table 5
IEA metrics on T_{BASE} and T_{EXTRA} test sets.

Info.	$\Delta C_{Point}(cm)$	$\Delta V_{Magnitude}(cm)$	$\Delta V_{Direction}(°)$
Pipe	-	1.94	4.20
Elbow	2.70	-	-
Conn.	1.69	-	-
Valve	1.63	-	13.31

Table 6
IUA metrics on T_{MAP} test set.

Info.	$\Delta C_{Point}(cm)$	$\Delta V_{Magnitude}(cm)$	$\Delta V_{Direction}(°)$
Pipe	-	3.12	2.42
Elbow	3.40	-	-
Conn.	-	-	-
Valve	3.59	-	13.92

environment conditions like the ones present in the T_{EXTRA} test set.

The metrics presented in this section outperform other state of the art methods for pipe recognition: [26] - traditional computer vision algorithms over 2D underwater images achieving an F1-score of 94,1%, [32] - traditional computer vision algorithms over 2D underwater images achieving a mean F1-score over three datasets of 88.0% and [17] - deep leaning approach for 2D drone imagery achieving a pixel-wise accuracy of 73.1%.

For the valve recognition, the only works that consider this class are [21, 51]. The metrics presented in these works are not comparable as their main focus is the classification of different types of valves without providing information about detection rates.

4.2. *IEA* Results

Table 5 shows the evaluation metrics obtained by the *IEA* when applied over the network segmentation of the T_{BASE} and T_{EXTRA} test sets altogether.

The *IEA* was able to detect all pipes, elbows, connections and valves present in the network segmentation, without generating any false positives.

For the pipe information, the vector magnitude (pipe length) and direction (pipe orientation) are evaluated. The difference between *IEA* output and ground truth for the magnitude is $1.94cm$, and 4.2° for the direction. For the elbow and connection information, only the central point position is evaluated, for which the differences are $2.7cm$ and $1.69cm$, respectively. With these metrics, it can be determined that the *IEA* is able to determine pipe length, orientation and its different structural elements with high accuracy.

For the valve information, the central point (valve position) and its vector direction (valve orientation) are evaluated, obtaining a divergence of $1.63cm$ and 13.31°, respectively. Furthermore, the valve type is correctly identified 73.6% of the time. Even though the valve position is detected with high accuracy, it exists a small error when determining its orientation and a higher one when classifying its type.

Qualitative results of the neural network segmentation and *IEA* output over diverse point clouds are shown in Figure 12.

4.3. *IUA* Results

Table 6 shows the evaluation metrics obtained after applying the *IUA* on the information extracted by the *IEA* from the network segmentation of the T_{MAP} test set. For this execution, the validation count check was executed each five analysed point cloud information ($K = 5$) with a count threshold of 2.

All pipes, elbows and valves were detected without generating any false positives. Additionally, the presented metrics are very similar to the ones obtained by the *IEA* execution on the T_{BASE} and T_{EXTRA} test sets, which implies that the *IUA* is able to merge the information from diverse point clouds while preserving its quality.

Figure 13 shows the T_{MAP} test set original point clouds along the *IEA* information extraction output and the final unified information by the *IUA*.

5. *AUV* Online Implementation

An objective of this work is to implement the semantic segmentation network and information algorithms on an AUV and execute them online during manipulation and inspection tasks. This section describes the used *AUV* characteristics, the online implementation of the neural network and information algorithms, and its validation.

5.1. *AUV* Description

The used *AUV* is a SPARUS II model unit [6] (Figure 14) equipped with three motors, granting it three degrees of mobility (surge, heave and yaw). Its navigation payload is composed by: 1) a Doppler Velocity Logger (*DVL*) to get linear and angular speeds and altitude, 2) a pressure sensor which provides depth measurements, 3) an Inertial Measurement Unit (*IMU*) to measure accelerations and angular speeds, 4) a Compass for heading, 5) a GPS to be georeferenced during surface navigation, and 6) a Short Baseline acoustic Link (*USBL*) used for localisation and data exchange between the robot and a remote station. Additionally, it has installed a stereo pair of Manta G283 cameras facing downwards.

The robot has two computers. One is dedicated to receive and manage the navigation sensor data and run the main robot architecture, developed under *ROS* (Intel i7 processor at 2.2 GHz, Intel HD Graphics 3000 engine and 4 GB of RAM). The second computer is used to capture the images from the stereo cameras and execute the online semantic segmentation and information algorithms (Intel i7 processor at 2.5 GHz, Intel Iris Graphics 6100 and 16GB of RAM).

The localisation of the vehicle is obtained through the fusion of multiple state estimations produced by the *DVL*, *IMU*, Compass, GPS, *USBL*, visual odometry and a navigation filter [14]. This localisation can be integrated into the point clouds generated from the images captured by the

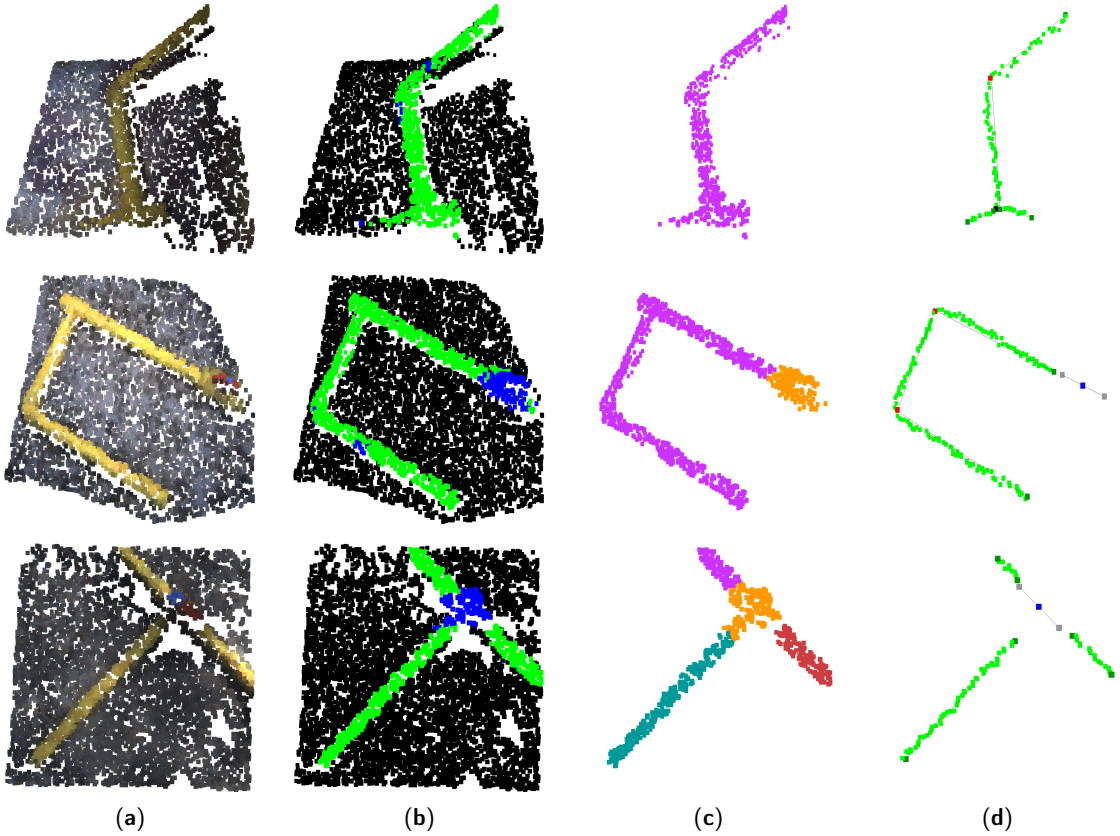


Figure 12: Neural network segmentation and *IEA* qualitative results. **(a)** Original point cloud. **(b)** Deep neural network segmentation. **(c)** Instance clustering. **(d)** Information extracted from *IEA*.

stereo pair of cameras to spatially reference them, which is a requirement to execute the *IUA*.

5.2. Implementation

To perform the online implementation we design a pipeline based on *ROS*.

The first step is to transform the images published from the stereo pair into point clouds to be processed by the neural network. To do so, diverse C++ *ROS* nodes are set up to: 1) rectify the raw images using the camera calibration parameters, 2) decimate the rectified images from their original size ($1920 \times 1440 pixels$) to $960 \times 720 pixels$, 3) calculate the disparity map and generate the point clouds and 4) downsample the point clouds using a voxel grid. Additionally, a python *ROS* node is set up to subscribe to the downsampled point clouds.

Following, the point cloud is fed into a previously loaded inference graph of a *DGCNN* trained model, performing the semantic segmentation. From there, the *IEA* and *IUA* can be executed. Finally, a publishing python *ROS* node is set up to publish the extracted information back into *ROS* to be accessed by other robots, sensors or actuators.

5.3. Validation

To validate the online execution, the frame rate of the output information stream is evaluated. An online execution was performed during the immersions conforming the $S_{POOL\text{-}2}$ and $S_{SEA\text{-}3}$ sets. In total, the online workflow was tested for 15'23". For each execution the achieved output information stream frame rate and the time that each online execution step described in 5.2 takes are calculated.

For each immersion, the inspected pipe and valve configuration are different, making the *IEA* and *IUA* algorithms execution time vary, as the number and shape of pipe and valves is different, making the time analysis more robust as it covers a wider variety of scenarios.

The average output information stream frame rate and times for each online execution step are calculated as the mean value from both executions. Figure 15 presents a breakdown of the total average online execution time into its different steps.

The total average online execution time is 1.39 $seconds$, which results in an output information stream frame rate of $0.72 fps$. The preprocessing step takes a mean of $68ms$ (4.9% of the total time) and includes all operations to transform the images published from the stereo pair into point clouds to be processed by the neural network. The network inference takes the biggest amount of time with a mean of $690ms$ (49.8% of the total time). Following, the *IEA* and *IUA* take a mean of $411ms$ and $210ms$, accounting for 29.7% and 15.1% of the total time, respectively. Finally, the information publication takes a mean of $7ms$ (0.5% of the total time).

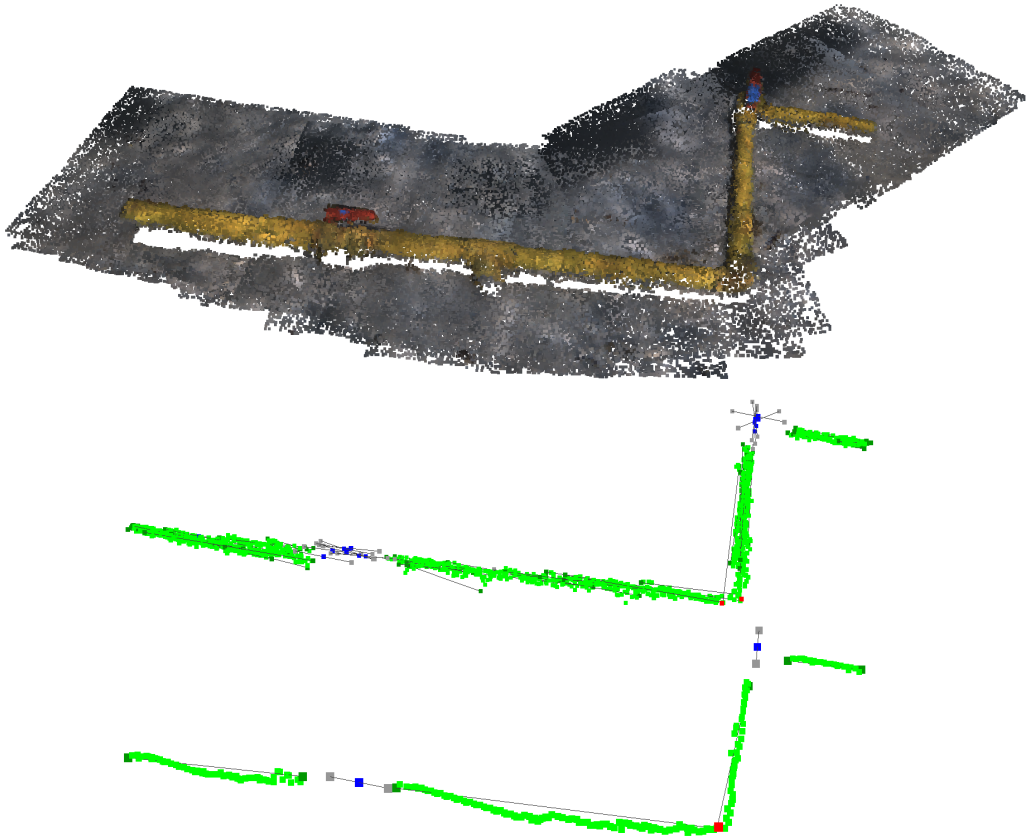


Figure 13: *IUA* implementation over T_{MAP} test set. Top: Overlapped point clouds. Middle: Overlapped extracted information from *IEA*. Bottom: Unified information from *IUA*.

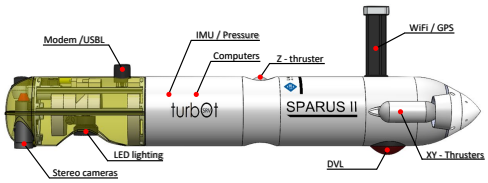


Figure 14: SPARUS II AUV.

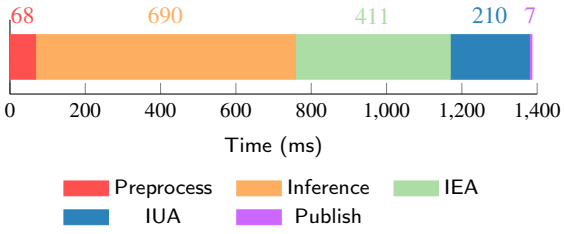


Figure 15: Online execution time breakdown.

The achieved output information stream frame rate is more than enough to perform manipulation tasks, as this kind of operations in underwater scenarios tend to have slow and controlled dynamics. Additionally, for most manipulations tasks the *IUA* may not be executed, lowering the overall online execution time to 1.18 $seconds$, and thus increasing the achieved frame rate to 0.85 fps.

Regarding inspections tasks, a way to validate the achieved output information stream frame rate is to check if exists an overlap between the analysed point clouds, ensuring full coverage of the inspected area. To do so, overlap between the original images from analysed point clouds is checked. This overlap depends on the camera displacement between the images from two consecutive analysed point clouds (d_{KF}) and on the height of the image footprint (h_{FP}). Then, the *overlap* can be expressed as:

$$overlap = (h_{FP} - d_{KF}) \cdot h_{FP}^{-1} , \tag{6}$$

$$d_{KF} = v \cdot frame\_rate^{-1} , \tag{7}$$

$$h_{FP} = (a \cdot h_{image}) \cdot f^{-1} . \tag{8}$$

Where v denotes the AUV velocity, a the navigation altitude, h_{image} the image height pixels and f the camera focal length.

During inspection tasks, an *AUV* like the SPARUS II can achieve velocities up to $v = 0.4 m/s$ and navigate at

a minimum altitude of $a = 1.5m$. Using these parameters along the Manta G283 camera intrinsic focal length of $f = 1505.5p$ and image height resolution of $h_{image} = 1440p$, the obtained overlap is 61.4%. Thus, the output information stream frame rate is high enough to get point clouds to overlap even in the most adverse navigation conditions.

6. Conclusions and Future Work

This paper presented the implementation of the *DGCNN* deep neural network to perform pixel-wise 3D segmentation of underwater pipes and valves from point clouds. To train the network, multiple immersions were conducted with an *AUV* to gather point clouds containing diverse pipe and valve configurations.

Two information algorithms have been developed, the first one groups the segmented pixels into instances and implements an object detection, detecting pipe and valve instances in a point cloud. Following, it extracts information from the detected instances providing pipe vectors, gripping points, structural elements like elbows or connections, and valve type and orientation. The second algorithm unifies information from spatially referenced point clouds, forming an information map of an inspected area.

Lastly, a *ROS* pipeline is build to execute the 3D segmentation and information extraction and unification algorithms online on an *AUV*, performing real-time underwater pipe and valve recognition, characterisation and mapping for inspection and manipulation tasks.

The neural network evaluation presented good results, reaching a mean *F1-score* value of 88.0% between the two conducted tests at a pixel-wise level and of 95.3% at an instance-level. Validating the usage of the *DGCNN* deep neural network on underwater scenarios.

The information extraction algorithm results showcased excellent metrics when extracting information from pipe instances and its structural elements (elbows and connections), and good metrics when extracting valves position, orientation and type. The mapping algorithm was able to merge information from diverse point clouds, preserving its quality and deleting spurious false positive detections.

Finally, the online execution validation demonstrated that the output information stream frame rate is high enough to perform manipulation tasks and to get point clouds to overlap, permitting an adequate implementation of the information unification algorithm and ensuring a full area coverage during inspection tasks.

It is important to point out that the whole workflow presented in this work is executed using a simple point cloud as an input, no matter what its source is (i.e. stereo vision, sonar, laser, ...). Thus, it can be implemented and utilised in multiple scenarios covering a wide range of applications.

Further developments will focus on studying the implementation of new deep neural networks to improve its segmentation performance and reduce the inference time, as this is the most time consuming step of the online execution. Additionally, new ways of extracting pipe and valve information will be studied to improve pose and type detection [20], maybe even being able to detect the valve handle position, providing the valve state and allowing the generation of pipeline flow diagrams.

Acknowledgments

Miguel Martin-Abadal was supported by Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contract DPI2017-86372-C3-3-R. Gabriel Oliver-Codina was supported by Ministerio de Ciencia e Innovacion, Agencia Estatal de Investigación (MCIN,AEI), under grant PID2020-115332RB-C33 10.13039/501100011033 and, as appropriate, by "ERDF A way of making Europe" . Yolanda Gonzalez-Cid was supported by Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contracts TIN2017-85572-P and DPI2017-86372-C3-3-R; and by the Comunitat Autonoma de les Illes Balears through the Direcció General de Política Universitaria i Recerca with funds from the Tourist Stay Tax Law (PRD2018/34).

CRediT authorship contribution statement

Miguel Martin-Abadal: Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. **Gabriel Oliver-Codina:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Yolanda Gonzalez-Cid:** Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

[1] Ahmed, S., Khan, M.F.R., Labib, M.F.A., Chowdhury, A.E., 2020. An observation of vision based underwater object detection and tracking, in: 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), pp. 117–122. doi:doi: 10.1109/ICETCE48199.2020.9091752.

[2] Asakawa, K., Kojima, J., Kato, Y., Matsumoto, S., Kato, N., 2000. Autonomous underwater vehicle aqua explorer 2 for inspection of underwater cables, in: Proceedings of the 2000 International Symposium on Underwater Technology (Cat. No.00EX418), pp. 242–247. doi:doi: 10.1109/UT.2000.852551.

[3] Bazeille, S., Quidu, I., Jaulin, L., 2012. Color-based underwater object recognition using water light attenuation. Intelligent Service Robotics 5, 109–118. doi:doi: 10.1007/s11370-012-0105-3.

[4] Burguera, A., Bonin-Font, F., 2020. On-line multi-class segmentation of side-scan sonar imagery using an autonomous underwater vehicle. Journal of Marine Science and Engineering 8, 557. doi:doi: 10.3390/jmse8080557.

[5] Capocci, R., Dooly, G., Omerdić, E., Coleman, J., Newe, T., Toal, D., 2017. Inspection-class remotely operated vehicles—a review. Journal of Marine Science and Engineering 5, 13. doi:doi: 10.3390/jmse5010013.

[6] Carreras, M., Hernández, J.D., Vidal, E., Palomeras, N., Ribas, D., Ridao, P., 2018. Sparus II AUV - A hovering vehicle for seabed inspection. IEEE Journal of Oceanic Engineering 43, 344–355. doi:doi: 10.1109/JOE.2018.2792278.

[7] Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85. doi:doi: 10.1109/CVPR.2017.16.

[8] Chen, Z., Wang, H., Xu, L., Shen, J., 2014. Visual-adaptation-mechanism based underwater object extraction. Optics and Laser Technology 56, 119–130. doi:doi: 10.1016/j.optlastec.2013.07.003.

[9] Cheng, J.C., Wang, M., 2018. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. Automation in Construction 95, 155–171. doi:doi: 10.1016/j.autcon.2018.08.006.

[10] Costa, M., Pinto, J., Ribeiro, M., Lima, K., Monteiro, A., Kowalczyk, P., Sousa, J., 2019. Underwater archaeology with light auvs, in: OCEANS 2019 - Marseille, pp. 1–6. doi:doi: 10.1109/OCEANSE.2019.8867503.

[11] Denos, K., Ravaut, M., Fagette, A., Lim, H., 2017. Deep learning applied to underwater mine warfare, in: OCEANS 2017 - Aberdeen, pp. 1–7. doi:doi: 10.1109/OCEANSE.2017.8084910.

[12] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The PASCAL Visual Object Classes (VOC) challenge. International Journal of Computer Vision 88, 303–338. doi:doi: 10.1007/s11263-009-0275-4.

[13] Fatan, M., Daliri, M.R., Mohammad Shahri, A., 2016. Underwater cable detection in the images using edge classification based on texture information. Measurement: Journal of the International Measurement Confederation 91, 309–317. doi:doi: 10.1016/j.measurement.2016.05.030.

[14] Font, E.G., Bonin-Font, F., Negre, P.L., Massot, M., Oliver, G., 2017. USBL Integration and Assessment in a Multisensor Navigation Approach for field AUVs. International Federation of Automatic ControlIFAC (IFAC) 50, 7905–7910. doi:doi: 10.1016/j.ifacol.2017.08.754.

[15] Foresti, G.L., Gentili, S., 2002. A hierarchical classification system for object recognition in underwater environments. IEEE Journal of Oceanic Engineering 27, 66–78. doi:doi: 10.1109/48.989889.

[16] Geisser, S., 1975. The predictive sample reuse method with applications. Journal of the American Statistical Association 70, 320–328. doi:doi: 10.1080/01621459.1975.10479865.

[17] Guerra, E., Palacin, J., Wang, Z., Grau, A., 2020. Deep learning-based detection of pipes in industrial environments, in: Industrial Robotics - New Paradigms. IntechOpen. doi:doi: 10.5772/intechopen.93164.

[18] Heshmati-Alamdari, S., Bechlioulis, C.P., Karras, G.C., Nikou, A., Dimarogonas, D.V., Kyriakopoulos, K.J., 2018. A robust interaction control approach for underwater vehicle manipulator systems. Annual Reviews in Control 46, 315 – 325. doi:doi: https://doi.org/10.1016/j.arcontrol.2018.10.003.

[19] Heshmati-Alamdari, S., Nikou, A., Dimarogonas, D.V., 2021. Robust trajectory tracking control for underactuated autonomous underwater vehicles in uncertain environments. IEEE Transactions on Automation Science and Engineering 18, 1288–1301. doi:doi: 10.1109/TASE.2020.3001183.

[20] Himri, K., Pi, R., Ridao, P., Gracias, N., Palomer, A., Palomeras, N., 2018. Object recognition and pose estimation using laser scans for advanced underwater manipulation, in: 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), pp. 1–6. doi:doi: 10.1109/AUV.2018.8729742.

[21] Himri, K., Ridao, P., Gracias, N., 2021. Underwater object recognition using point-features, bayesian estimation and semantic information. Sensors 21. URL: https://www.mdpi.com/1424-8220/21/5/1807, doi:doi: 10.3390/s21051807.

[22] Jacobi, M., Karimanzira, D., 2013. Underwater pipeline and cable inspection using autonomous underwater vehicles, in: 2013 MTS/IEEE OCEANS - Bergen, pp. 1–6. doi:doi: 10.1109/OCEANS-Bergen.2013.6608089.

[23] Jalal, A., Salman, A., Mian, A., Shortis, M., Shafait, F., 2020. Fish detection and species classification in underwater environments using deep learning with temporal information. Ecological Informatics 57, 101088. doi:doi: 10.1016/j.ecoinf.2020.101088.

[24] Jeon, M., Lee, Y., Shin, Y.S., Jang, H., Kim, A., 2019. Underwater Object Detection and Pose Estimation using Deep Learning. IFAC-PapersOnLine 52, 78–81. doi:doi: 10.1016/j.ifacol.2019.12.286.

[25] Jonsson, P., Sillitoe, I., Dushaw, B., Nystuen, J., Heltne, J., 2009. Observing using sound and light – a short review of underwater acoustic and video-based methods. Ocean Science Discussions 6, 819–870. doi:doi: 10.5194/osd-6-819-2009.

[26] Kallasi, F., Oleari, F., Bottioni, M., Lodi Rizzini, D., Caselli, S., 2014. Object detection and pose estimation algorithms for underwater manipulation, in: 2014 Conference on Advances in Marine Robotics Applications.

[27] Kim, B., Yu, S., 2017. Imaging sonar based real-time underwater object detection utilizing adaboost method, in: 2017 IEEE Underwater Technology (UT), pp. 1–5. doi:doi: 10.1109/UT.2017.7890300.

[28] Kim, D., Lee, D., Myung, H., Choi, H., 2012. Object detection and tracking for autonomous underwater robots using weighted template matching, in: 2012 Oceans - Yeosu, pp. 1–5. doi:doi: 10.1109/OCEANS-Yeosu.2012.6263501.

[29] Kumar, S.S., Abraham, D.M., Jahanshahi, M.R., Iseley, T., Starr, J., 2018. Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks. Automation in Construction 91, 273–283. doi:doi: 10.1016/j.autcon.2018.03.028.

[30] Lee, D., Kim, G., Kim, D., Myung, H., Choi, H.T., 2012. Vision-based object detection and tracking for autonomous navigation of underwater robots. Ocean Engineering 48, 59–68. doi:doi: 10.1016/j.oceaneng.2012.04.006.

[31] Lee, S., Park, B., Kim, A., 2019. A deep learning based submerged body classification using underwater imaging sonar, in: 2019 16th International Conference on Ubiquitous Robots (UR), pp. 106–112. doi:doi: 10.1109/URAI.2019.8768581.

[32] Lodi Rizzini, D., Kallasi, F., Aleotti, J., Oleari, F., Caselli, S., 2017. Integration of a stereo vision system into an autonomous underwater vehicle for pipe manipulation tasks. Computers and Electrical Engineering 58, 560–571. doi:doi: 10.1016/j.compeleceng.2016.08.023.

[33] Lodi Rizzini, D., Kallasi, F., Oleari, F., Caselli, S., 2015. Investigation of vision-based underwater object detection with multiple datasets. International Journal of Advanced Robotic Systems 12, 1–13. doi:doi: 10.5772/60526.

[34] Martin-Abadal, M., Oliver-Codina, G., Gonzalez-Cid, Y., 2021a. Project webpage for "real-time pipe and valve characterisation and mapping for autonomous underwater intervention tasks". http://srv.uib.es/3d-pipes-2/. Accessed: Oct. 2021.

[35] Martin-Abadal, M., Piñar-Molina, M., Martorell-Torres, A., Oliver-Codina, G., Gonzalez-Cid, Y., 2021b. Underwater pipe and valve 3d recognition using deep learning segmentation. Journal of Marine Science and Engineering 9. doi:doi: 10.3390/jmse9010005.

[36] Martin-Abadal, M., Ruiz-Frau, A., Hinz, H., Gonzalez-Cid, Y., 2020. Jellytoring: Real-time jellyfish monitoring based on deep learning object detection. Sensors 20, 1–21. doi:doi: 10.3390/s20061708.

[37] Narimani, M., Nazem, S., Loueipour, M., 2009. Robotics vision-based system for an underwater pipeline and cable tracker, in: OCEANS 2009-EUROPE, pp. 1–6. doi:doi: 10.1109/OCEANSE.2009.5278327.

[38] Nikou, A., Verginis, C.K., Dimarogonas, D.V., 2018. A tube-based mpc scheme for interaction control of underwater vehicle manipulator systems, in: 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), pp. 1–6. doi:doi: 10.1109/AUV.2018.8729801.

[39] Nunes, A., Gaspar, A.R., Matos, A., 2019. Critical object recognition in underwater environment, in: OCEANS 2019 - Marseille, pp. 1–6. doi:doi: 10.1109/OCEANSE.2019.8867360.

[40] Olmos, A., Trucco, E., 2002. Detecting man-made objects in unconstrained subsea videos, in: British Machine Vision Conference, pp. 50.1–50.10. doi:doi: 10.5244/C.16.50.

[41] Ortiz, A., Simó, M., Oliver, G., 2002. A vision system for an underwater cable tracker. Machine Vision and Applications 13, 129–140. doi:doi: 10.1007/s001380100065.

[42] Palomer, A., Ridao, P., Youakim, D., Ribas, D., Forest, J., Petillot, Y., 2018. 3D laser scanner for underwater manipulation. Sensors 18, 1–18. doi:doi: 10.3390/s18041086.

[43] Prats, M., García, J.C., Wirth, S., Ribas, D., Sanz, P.J., Ridao, P., Gracias, N., Oliver, G., 2012. Multipurpose autonomous underwater intervention: A systems integration perspective, in: 2012 20th Mediterranean Conference on Control Automation (MED), pp. 1379–1384. doi:doi: 10.1109/MED.2012.6265831.

[44] Prechelt, L., 2012. Early Stopping — But When?. Springer. pp. 53–67. doi:doi: 10.1007/978-3-642-35289-8_5.

[45] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A., 2009. Ros: an open-source robot operating system, in: ICRA Workshop on Open Source Software.

[46] Rayhana, R., Jiao, Y., Liu, Z., Wu, A., Kong, X., 2020. Water pipe valve detection by using deep neural networks, in: Smart Structures and NDE for Industry 4.0, Smart Cities, and Energy Systems, SPIE. pp. 20 – 27. doi:doi: 10.1117/12.2558886.

[47] Rekik, F., Ayedi, W., Jallouli, M., 2018. A trainable system for underwater pipe detection. Pattern Recognition and Image Analysis 28, 525–536. doi:doi: 10.1134/S1054661818030185.

[48] Ridao, P., Carreras, M., Ribas, D., Sanz, P.J., Oliver, G., 2015. Intervention auvs: The next challenge. Annual Reviews in Control 40, 227–241. doi:doi: 10.1016/j.arcontrol.2015.09.015.

[49] Stereo Proc Repository, . Ros - stereo image proc. http://wiki.ros.org/stereo_image_proc. Accessed: 2022-05-18.

[50] Tascini, G., Zingaretti, P., Conte, G., 1996. Real-time inspection by submarine images. Journal of Electronic Imaging 5, 432– 442. doi:doi: 10.1117/12.245766.

[51] Vallicrosa, G., Himri, K., Ridao, P., Gracias, N., 2021. Semantic mapping for autonomous subsea intervention. Sensors 21. URL: https://www.mdpi.com/1424-8220/21/20/6740, doi:doi: 10.3390/s21206740.

[52] Wang, X., Liu, S., Liu, Z., 2017. Underwater sonar image detection: A combination of nonlocal spatial information and quantum-inspired shuoed frog leaping algorithm. PLoS ONE 12, 1–30. doi:doi: 10.1371/journal.pone.0177666.

[53] Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic graph cnn for learning on point clouds. ACM Trans. Graph. 38. doi:doi: 10.1145/3326362.

[54] Yu, M., Ariamuthu Venkidasalapathy, J., Shen, Y., Quddus, N., Mannan, M.S., 2017. Bow-tie analysis of underwater robots in offshore oil and gas operations, in: Offshore Technology Conference. doi:doi: 10.4043/27818-MS.

[55] Zingaretti, P., Zanoli, S.M., 1998. Robust real-time detection of an underwater pipeline. Engineering Applications of Artificial Intelligence 11, 257–268. doi:doi: 10.1016/S0952-1976(97)00001-8.

Chapter 4

Jellyfish detection and quantification

This chapter presents the work carried out on jellyfish control.

Jellyfish have been recognised as an important part of marine ecosystems, providing multiple benefits (Hays, Doyle, and Houghton, 2018; Lamb et al., 2019). Recently, an increase in its numbers has been linked to global change scenarios such as high fishing pressure (Richardson et al., 2009) and global warming (Brotz et al., 2012). This increase can create a multitude of impacts on human wellbeing, such as clogging seawater intake systems in water desalination and power plants (Lee et al., 2006), killing farmed fish in pens (Purcell, Baxter, and Fuentes, 2013) or creating negative impacts on coastal tourism (Fenner, Lippmann, and Gershwin, 2010).

Jellyfish monitoring efforts using underwater video observations tend to have limited spatial and temporal coverage due to human-based data logging approaches ranging from quantitative to presence/absence and relative abundance indices (Condon et al., 2013). The scarcity of consistent long-term temporal and spatial data on jellyfish is such that there is uncertainty about its population growth (Pitt et al., 2018).

The objective of this work is to develop a tool that can automatically detect and quantify different species of jellyfish based on a deep object detection neural network, recording jellyfish presence over long periods.

The first step was to collect the required data. Hundreds of images containing three species of jellyfish were gathered from publicly available videos on diverse social media sites. Next, an object detection CNN architecture was trained, and the best hyperparameters were selected. Then, a quantification algorithm was developed to track jellyfish occurrence on video recordings. Finally, the neural network and quantification algorithms were adapted to be executed online on stationary marine buoys, being able to log the presence of jellyfish in real-time.

This work is presented in detail in a journal article describing the data collection, network and hyperparameter selection and validation, quantification algorithms, and online implementation.

Title: Jellytoring: Real-Time Jellyfish Monitoring Based on Deep Learning Object Detection
Authors: **M. Martin-Abadal**, A. Ruiz-Frau, H. Hinz and Y. Gonzalez-Cid
Journal: Sensors
Published: 19 March 2020
Quality index: JCR2020 *Engineering, electrical & electronic*, IF 3.735, Q2 (82/273)

Jellytoring: Real-Time Jellyfish Monitoring Based on Deep Learning Object Detection

Miguel Martin-Abadal[a,*], Ana Ruiz-Frau[b], Hilmar Hinz[b] and Yolanda Gonzalez-Cid[a]

[a]Department of Mathematics and Computer Science. University of the Balearic Islands, 07122, Palma, Spain
[a]Department of Marine Ecosystem Dynamics, IMEDEA (CSIC-UIB), Institut Mediterrani d'Estudis Avançats, 07190, Esporles, Spain

ARTICLE INFO

The work presented in this preprint has been published in the journal *Sensors* as:

Martin-Abadal, M.; Ruiz-Frau, A.; Hinz, H.; Gonzalez-Cid, Y. Jellytoring: *Real-Time Jellyfish Monitoring Based on Deep Learning Object Detection*. Sensors 2020, 20, 1708.
DOI: 10.3390/s20061708

ABSTRACT

During the past decades, the composition and distribution of marine species have changed due to multiple anthropogenic pressures. Monitoring these changes in a cost-effective manner is of high relevance to assess the environmental status and evaluate the effectiveness of management measures. In particular, recent studies point to a rise of jellyfish populations on a global scale, negatively affecting diverse marine sectors like commercial fishing or the tourism industry. Past monitoring efforts using underwater video observations tended to be time-consuming and costly due to human-based data processing. In this paper, we present Jellytoring, a system to automatically detect and quantify different species of jellyfish based on a deep object detection neural network, allowing us to automatically record jellyfish presence during long periods of time. Jellytoring demonstrates outstanding performance on the jellyfish detection task, reaching an *F1 score* of 95.2%; and also on the jellyfish quantification task, as it correctly quantifies the number and class of jellyfish on a real-time processed video sequence up to a 93.8% of its duration. The results of this study are encouraging and provide the means towards an efficient way to monitor jellyfish, which can be used for the development of a jellyfish early-warning system, providing highly valuable information for marine biologists and contributing to the reduction of jellyfish impacts on humans.

1. Introduction

During the past decades, the marine environment has been under increased pressure by human activities, such as the over-exploitation of marine species [55], the destruction and modifications of habitats [34], the introduction of alien species [18], as well as pollution [33] and human-induced climate change [32, 35]. These pressures have caused highly relevant changes in the composition and distribution of marine organisms [26].

The detection and quantification of changes in marine species are of vital importance to monitor environmental status and its change over time, in particular, the benefits society derives from ecosystems, known as ecosystem services [50]. Furthermore, the capacity to monitor is critical in the assessment of the effectiveness of control or recovery measures implemented through management.

Visual observations of marine organisms using video cameras are increasingly adopted to monitor the marine environment due to the low cost of this technology and the wide applicability within a challenging environment for humans. Until recently, video observations have been processed and classified by human observers, which in many instances is time-consuming and consequently financially costly [6, 10].

In addition, the underwater environment is a highly dynamic environment, where a wide range of variables such as water turbidity, scale deformations, illumination variations, presence of flares, color distortions or light can affect the quality of the images collected, making data extraction a challenging undertaking.

*Corresponding author
miguel.martin@uib.es (M. Martin-Abadal)

Over the last decade, automatic detection methods have arisen as a cost-effective way for image location and classification [51], this is highly relevant in regards to the increasing amount of image data that is being collected from the marine environment. In general, images of animal species are used to record and quantify their density, distribution and behaviour [4, 22, 30, 68]. Getting to determine where objects are located in a given image (object localization) and which category each object belongs to (object classification) can be useful in a multitude of scenarios and implemented for multiple applications. In the marine environment object detection and classification has been used among others to record fish presence and recognition [39, 40, 69, 72], to monitor marine turtles [25] or in the classification of planktonic organisms [59].

General existing solutions for organisms automatic detection can be roughly classified into two groups: traditional computer vision algorithms or artificial intelligence based approaches.

Traditional computer vision algorithms use feature detection algorithms (SIFT, SURF, BRIEF, etc.) to extract feature information from the image (position of corners, edges, blobs, etc.). An object is recognized in a new image by individually comparing its features to a database and finding candidate matching features. The difficulty with these traditional approaches is the necessity to choose which features are important for each task. As the number of organisms to classify increases, feature extraction becomes more complex [53].

Artificial intelligence approaches, in turn, can be divided into two groups, machine learning and deep learning approaches:

Machine learning based approaches perform an informative region selection followed by a feature extraction of the selected regions (e.g., SIFT [45], HOG [9], Haar-like [41]) and finally a region classification using a determined method (e.g., Supported Vector Machine [7], AdaBoost [17], Deformable Part-based Model [14]). Still, the feature extraction process needs to be determined manually.

Deep learning based frameworks for image processing and object detection specifically, mostly rely on region-based *Convolutional Neural Networks* (CNN) like R-CNN [21] or its performance evolutions: Fast R-CNN [20] and Faster R-CNN [61], to generate deeper neural networks with more layers, able to learn and extract more complex features. Here, the full process is automated, with no need of a previous feature extraction, as the network inputs an image and is able to extract its own features.

In this paper, we present Jellytoring, a system to automatically detect and quantify different species of jellyfish based on a deep object detection neural network. Within the context of human–environment interactions, jellyfish are organisms that can create a multitude of impacts on human wellbeing. Among others, the presence of jellyfish aggregations can clog seawater intake screens in water desalination and power plants, causing power reductions and shutdowns [38, 48], leaving entire populations without electrical supply. In aquaculture, large aggregations of jellyfish can cause important socio-economic impacts by killing farmed fish in pens [49, 57]. In commercial fishing, jellyfish can interfere with fishing operations by constituting a health hazard to fishermen when retrieving the nets, by splitting the fishing nets due to the weight of the jellyfish in the nets or by ruining the catch [58]. Additionally, jellyfish are known to create negative impacts on coastal tourism by generating unpleasant experiences among coastal users with associated impacts on tourism revenues and the tourism industry [15].

The development of an automatic jellyfish detection and identification system could contribute to the reduction of jellyfish impacts on humans, providing the means towards an effective acquisition of jellyfish presence surveillance data which could be used for the development of a jellyfish early-warning system. The nature and characteristics of jellyfish, however, are challenging aspects to overcome in the development of such a system. Jellyfish are often translucent organisms whose bodies can adopt significantly different configurations, due to the movement of their tentacles in relation to their main body structure, i.e., the bell or umbrella. These aspects, translucent nature and changing shapes, together with the added difficulties of object detection in underwater environments, represent challenging conditions for the development of a jellyfish detection system.

We focused on the North-Western Mediterranean sea, an area with a high human population and a popular tourism destination, where human–jellyfish interactions are frequent. Specifically, we studied three jellyfish species which are common during the summer months and which often cause undesired effects on tourism satisfaction, namely *Pelagia noctiluca*, *Cotylorhiza tuberculata* and *Rhizostoma pulmo*.

The remainder of this paper is structured as follows. Section 2 reviews related work on jellyfish detection, quantification and monitoring and highlights our main contributions. Section 3 describes the used neural network architecture and its training details. Section 4 describes the adopted methodology and materials used in this study. The experimental results are presented in Section 5. Finally, Section 6 presents the main conclusions and outlines future work.

2. Related Work and Contributions

This section briefly describes the existing related efforts on jellyfish detection and monitoring. The main contributions of this paper are highlighted at the end of this section.

2.1. State-Of-The-Art

During the last decades, the monitoring of jellyfish species has mostly been carried out manually, relying on human visual observations to detect, identify and quantify specimens; that is either by direct observations made in the field [56] or by using video recordings that subsequently needed manual analysis [24]. The use of aerial vehicles has also been adopted, to cover a larger study areas [31]. However in general, visual observations tend to be slow, labour and resource intensive, thus restricting the spatial and temporal extent of the studies [29, 37].

Some studies have used the aid of traditional computer vision techniques to automate the detection of jellyfish. Rife et al. [62] tested various image filtering techniques and segmentation algorithms to track deep-ocean jellyfish on conventional camera imagery. However, this implementation only considers a generic jellyfish class, not distinguishing between different species. Moreover, the selected combination of filtering and segmentation algorithm does not allow for a real-time tracking application.

As in many other research areas, the recent development of deep learning architectures has offered major improvements in accuracy for observational ecological studies [70], dealing at the same time with the spatial and temporal limitations of human visual observation [71]. Even so, the application of deep learning for jellyfish detection has been very limited. To our knowledge, only two peer-reviewed publications have focused on the subject.

Kim et al. [36] make use of an unspecified CNN along with collaborative filters to build a jellyfish recognition algorithm for sea surface imagery taken by an unmanned aerial vehicle. Similar to the studies mentioned above, this study does also not distinguish between different species of jellyfish. Furthermore, limiting image capture to the water surface underestimates jellyfish numbers, as jellyfish distribution is not limited to surface waters only and tend to occupy a large extent of the underlying water column.

French et al. [16] implement a 10-layer VGG-style CNN architecture to detect jellyfish in underwater sonar imagery, correctly classifying up to a 90% of the jellyfish for the test set. The use of sonar imagery presents some advantages, like the usability at deeper areas where light does not reach. On the other hand, it suffers from some drawbacks versus the

usage of normal camera imagery, like lower resolution or grey-scale coloring, complicating the detection task. This study did not differentiate between different jellyfish species.

Finally, we found that none of these works performed a jellyfish quantification to provide information of occurrences over time, nor used time series processing techniques to improve the detection rate that allowed for the implementation of a monitoring algorithm.

2.2. Main Contributions

The main contributions of this paper are composed of:

1. A real-time jellyfish monitoring system based on deep learning object detection named Jellytoring, which provides highly valuable information to biologists, ecologists and conservationists on the presence and occurrence of different species of jellyfish in an studied area.
2. The usage of a deep CNN, trained several times to fine tune its hyperparameters to detect and classify up to three different species of jellyfish on underwater images. We evaluated the network on a test set of images, comparing its results to other neural networks.
3. First system to achieve real-time automatic quantification and identification of different species of jellyfish. We designed and tested an algorithm that can be executed in real-time and uses the network detection to quantify and monitor jellyfish presence on video sequences.
4. The creation of a publicly available dataset used for the training and testing of the neural network and the quantification algorithm, containing the original images and corresponding annotations.

3. Deep Learning Approach

This section describes the framework and network selection process along with its architecture and training details.

3.1. Framework and Network Selection

There are several deep learning frameworks based on CNN that can be used to extract instance information from images. They go from the standard region proposal based object detection frameworks of Faster R-CNN [61] or some of its direct evolutions like FPN [42], mask R-CNN [27] or RFCN [8]; to regression-based ones like YOLO [60] or SSD [44]; or even more specific frameworks like deep salient object detection [1].

In our case, we aim to implement an object detection framework able to detect and classify up to three species of jellyfishes present in underwater images, with no need of obtaining the pixel-wise segmentation of the detected instances nor any extra feature that could slow the process. We wanted to ensure that the system is able to perform real-time quantification on a wide spectrum of setups, widening its applicability.

Taking into account both the computational cost along with the features of the aforementioned frameworks and the requirements of our application, we opted for the usage of the Faster R-CNN framework. This framework allow us to obtain the jellyfish instances bounding boxes and its classification, while balancing the detection performance and computational cost trade off by selecting an adequate deep learning architecture for this specific task.

Due to the slow movement of the jellyfish, an architecture with high detection performance, despite having a relatively high image analysis time is suitable. Therefore, based on the performance metrics provided by Google on tests [23] conducted for diverse object detection architectures over the COCO dataset [43], we selected the Faster R-CNN-based implementation of the Inception ResNet v2 [63] architecture. It uses a region proposal network to generate object position instances and then the Inception ResNet v2 to fine-tune these proposals and output a final prediction, presenting a two-stage detection framework.

Inception ResNet v2 is a very deep CNN with over 450 layers that can efficiently learn to identify objects in images, outputting instance bounding boxes and classifying them into one of the specified classes with a confidence percentage.

Selecting appropriate kernel sizes for the convolutional layers is a crucial aspect when detecting objects in an image, as the same object may show variations in shape and size. Larger kernels are preferred for the detection of bigger objects while smaller kernels are favored for smaller ones. To address this variation, Inception-ResNet V2 architecture performs multiple parallel convolutions using different kernel sizes, making the network "wider" rather than "deeper". The blocks of layers containing these convolutions are called Inception Modules [64].

The network also uses Residual Connections [28], through which the output of the convolution operation of the Inception Module is added to the input. This introduces shortcuts in the model resulting in more optimal and accurate networks. This architecture combines Inception Modules and Residual Connections which results in the Inception-ResNet modules. Figure 1 shows a compressed view of the whole Inception ResNet v2 architecture. More in-depth information about this architecture can be found in [63].

3.2. Training Details

The Inception-ResNet V2 architecture is trained by means of readjusting the kernel values in the convolutional layer filters, back-propagating the loss computed over the predictions obtained on the softmax layers.

Due to the high number of layers, the loss becomes small and insufficient to update the kernel values properly. To prevent the middle part of the network from "dying out" during the backpropagation process, an auxiliary classifier is applied at the output of the second block of Inception-ResNet modules. In this way, an auxiliary loss is computed and added to the prior one as shown in Equation (1).

$$Total\_loss = main\_loss + aux\_loss \times 0.3\ . \tag{1}$$

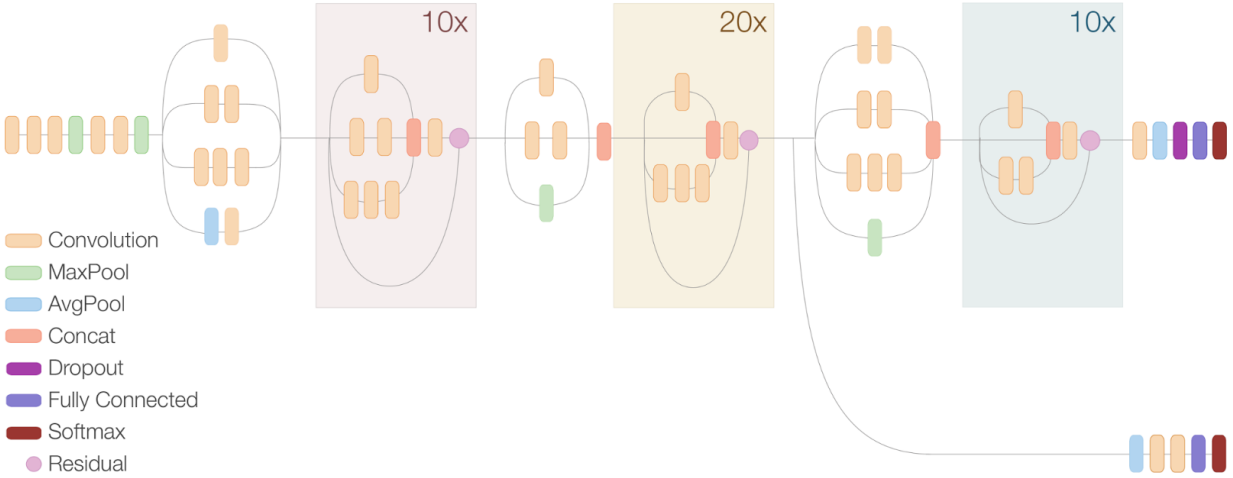


Figure 1: Inception ResNet v2 architecture. Credit: Google AI Blog.

To train the network and adjust the kernel weights, the smooth L1 location backpropagation loss function is used, which loss increases as the predicted bounding box location diverges from the ground truth. Additionally, the Momentum optimizer algorithm together with gradient clipping strategies [54] are utilized to achieve a minimum global error.

The architecture used for this application had already been trained over the COCO dataset [43]. To retrain the network, it is needed a set of images containing different species of jellyfish and their corresponding ground truth annotations, where the position and class of each jellyfish instance are indicated.

4. Methodology

This section introduces the general workflow of Jellytoring and subsequently provides details of each work step taken i.e., the acquisition and labeling of the data from the training and testing sets, the tested network hyperparameters and studied combinations, the validation process and evaluation metrics and finally, the quantification algorithm.

4.1. Workflow

First, a set of images containing jellyfish needs to be forwarded into a frozen version of a trained model of the deep object detection neural network. After its inference, the network generates the jellyfish detection.

Following, this detection is optimized by a *non-maxima suppression* (nms) algorithm [52], deleting overlapping ones. Then, the final predictions for each analyzed image are obtained by deleting instances with an associated confidence lower than a selected threshold value (C_{thr1}). These predictions can be used to measure jellyfish occurrences and species recognition in the forwarded images on its own.

Furthermore, if the initial source of data is a video sequence, the network detection can be forwarded into the quantification algorithm to obtain the evolution of number and species of jellyfish present on the video sequence. This algorithm first deletes instances with an associated confidence lower than a selected threshold value (C_{thr2}) and then applies time series processing techniques. More in depth information about the quantification algorithm is provided in Subsection 4.6.

Figure 2 represents the workflow of Jellytoring.

4.2. Data collection

The present study focuses on three jellyfish species, namely *Pelagia noctiluca*, *Rhizostoma pulmo* and *Cothyloriza tuberculata*. To obtain the needed data to train and test the neural network, we extracted images containing instances of the studied jellyfish from underwater video recordings.

The first source of data consisted of a series of recordings we generated by mounting a GoPro camera onto a platform and deploying it at the seafloor, pointing upwards. In order to obtain a variety of exposure conditions, recordings were done during different times of the day, over different seabed types and weather conditions. Using this method we generated up to 4 hours of recordings. Secondly, to obtain additional data, we examined diverse social media sites publicly available videos where appeared instances of the three studied jellyfish. From these sources, we extracted a total of 842 images, each one containing at least one jellyfish instance. When possible, images containing more than one instance were extracted. The resolution of the images range from $320x240$ to $1920x1080 pixels$, they can be forwarded into the network without any processing, as the network is able to process different image and bounding boxes sizes thanks to its multiple feature extraction kernels sizes and shapes.

We built a varied dataset containing jellyfish instances under different conditions, such as jellyfish coloration, position and size; or water illumination, depth and turbidity. We obtained a varied and robust dataset to train the neural network without overfitting the training data and to test it on different scenarios to ensure its wide usability. Figure 3

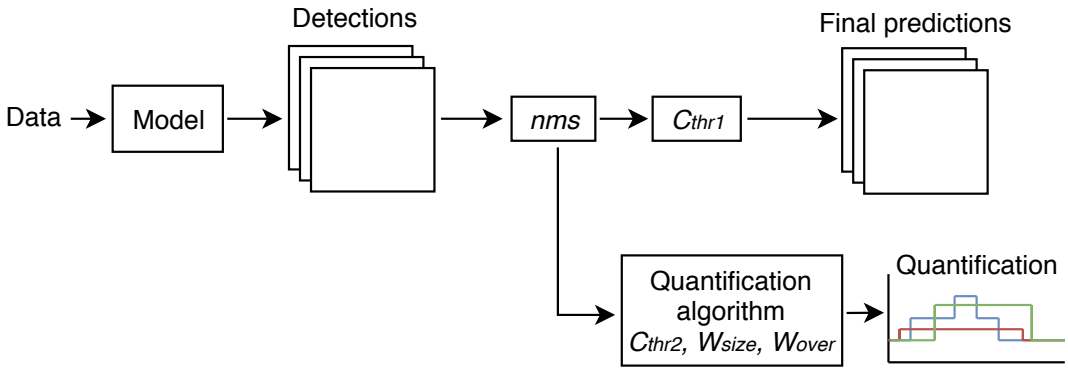


Figure 2: Jellytoring workflow.

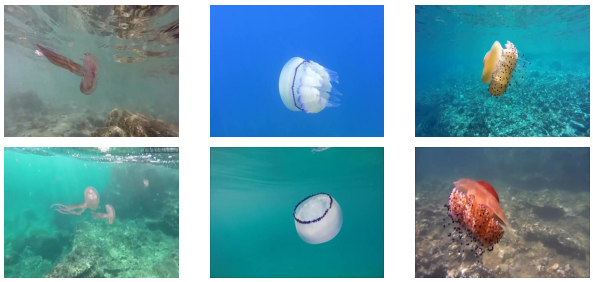


Figure 3: Images from the dataset showing the three jellyfish species under different environmental conditions. Left: *P. noctiluca*, center: *R. pulmo*, right: *C. tuberculata*.

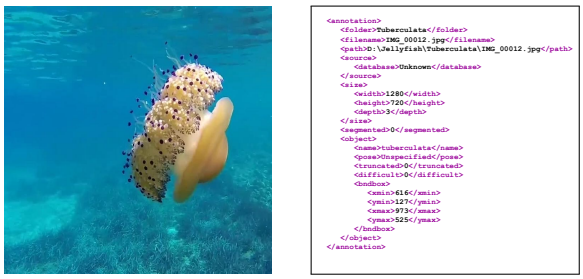


Figure 4: Left: Original image. Right: Corresponding ground truth ".xml" file, specifying the jellyfish location and class.

shows sample images from the dataset show-casing different environmental conditions.

To log the presence of the different jellyfish species, annotation files were generated using the LabelImg tool [67]. For each image, a bounding box around each jellyfish instance was drawn and was classified according to its species. The LabelImg tool then generates an ".xml" file containing the position and classification of each instance within the corresponding image. A total of 962 jellyfish occurrences were recorded, 327 corresponding to *Pelagia noctiluca*, 292 to *Rhizostoma pulmo* and 343 to *Cothyloriza tuberculata*. Figure 4 shows an original image along with its ground truth ".xml" text file.

Table 1
Hyperparameter values and combinations.

Index	Data aug.	Learn. rate	Iterations
1	No	5e-04	10k
2			20k
3			40k
4		decay	10k
5			20k
6			40k
7	Yes	5e-04	10k
8			20k
9			40k
10		decay	10k
11			20k
12			40k

4.3. Hyperparameter Selection

When training a neural network the value of specific hyperparameters can determine some of the network features and the training process itself. To find the values of these hyperparameters that offer the best performance, the network was trained using different values and combinations.

The considered hyperparameters were:

- Data augmentation: it is a technique that consists of applying random rotations and horizontal and vertical transformations to the training images in order to train over more diverse data, helping to reduce overfitting [66].
- Learning rate: this hyperparameter modifies the training step size the network uses when searching for an optimal solution. We also studied the effect of applying a decay learning rate, which consists of lowering the learning rate value as the training progresses [2].
- Number of iterations: this hyperparameter sets the number of times the network back-propagates and trains [2].

Table 1 shows the values and combinations of hyperparameters that we used to train the neural network.

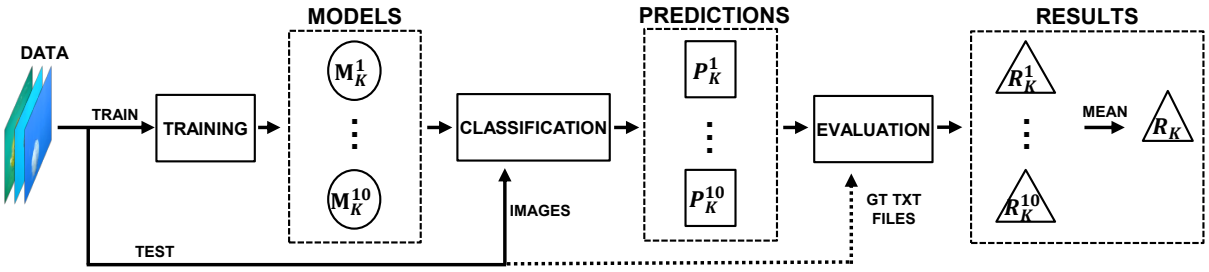


Figure 5: Experiment *K* validation process. For each one of the twelve hyperparameter combinations, the network was trained ten times using the k-fold cross-validation method, outputting ten models. These models were run and evaluated over the test data. Finally, the results of the models were obtained and its mean performance calculated.

4.4. Validation

We conducted twelve different experiments, each one assessing the performance of hyperparameter combination. When training the network, we made use of the 10k-fold cross-validation method [19]. Through this method, the dataset is split into ten equally sized subsets and the network is trained ten times, each time using two different subsets as the test data (20% of the dataset) and the remaining eight subsets as training data (80% of the dataset). This method reduces the variability of the results, as these are less dependent on the selected test and training datasets, therefore obtaining a more accurate performance estimation.

Using the 10k-fold cross-validation, ten models were generated for each experiment, M_K^i, where K=*1..12* represents the experiment number and i=*1..10* the model index. We ran the ten output models with their corresponding test subsets, obtaining jellyfish detection of all the models.

To remove overlapped detection and obtain the final predictions of each model, P_K^i, an nms algorithm is applied. This algorithm computes the intersection area between detection and eliminates the least confident ones when the intersection area is greater than a threshold. Threshold values for this type of application are usually set between 30%–70% [3, 5], in our case, we selected a fairly restrictive threshold of 40%, as it is not common that two or more jellyfish appear superimposed in the images.

From these predictions, each model is evaluated in terms of detection performance, obtaining its results metrics R_K^i. Finally, the detection performance R_K of each experiment is computed as the mean of its ten R_K^i models performance. The best model M corresponding to the experiment that presented the best results is selected to generate the quantification and monitoring predictions. The validation process of the experiments is shown in Figure 5.

4.5. Model Evaluation

The first step to evaluate a model and measure its performance is to classify each one of the predictions over the test set data as either correct (*True Positive*, TP) or incorrect (*False Positive*, FP). To do so, we used the *Intersection over Union* (IoU) measure, which provides the similarity between the predicted and the ground-truth bounding-boxes areas. The IoU value is defined as the area of the intersection between bounding-boxes divided by the union of the bounding-boxes areas (Equation (2)).

$$IoU = \frac{A_{intersection}}{A_{union}} . \tag{2}$$

To determine whether a prediction is a TP or an FP, an IoU threshold value needs to be established. Following the criteria applied in the PASCAL VOC challenge [12], this threshold was set at thr_{iou}= *0.5*. A prediction is classified as TP if the IoU value with any ground truth bounding-box is greater than the thr_{iou} and the predicted class matches the corresponding one specified in the ground truth box. Otherwise, the prediction is classified as an FP (Equation (3)).

$$Prediction = \begin{cases} TP, & \text{if } IoU >= thr_{iou} \ \& \ Class_{pred} = Class_{gt} \\ FP, & \text{otherwise} . \end{cases} , \tag{3}$$

Finally, ground-truth instances that do not have a IoU>thr_{iou} with any prediction are counted as undetected instances (*False Negatives*, FN).

Once each prediction is classified as either TP or FP, and the number of FN is obtained, evaluation metrics are computed.

Average Precision (AP) [73] is one the most frequently used metrics in object detection applications. It is largely used in object detection competitions such as PASCAL VOC [12], ImageNet [11] or COCO [43]. This metric takes into account all predictions, offering a solid comparative standard between networks and applications. Once the AP is obtained for each class, a *mean Average Precision* (mAP) for all classes is computed.

Following, a threshold sweep over the prediction confidence from 0% to 100% in 1% steps was performed (C_{thr1}). For each step, the predictions with an associated confidence level lower than the C_{thr1} were removed; and the *Precision* and *Recall* metrics from the TP, FP and FN values were calculated.

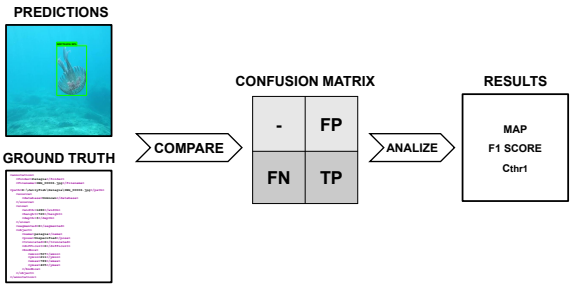


Figure 6: Model evaluation process. The final predictions are compared to their corresponding ground truth using the Intersection over Union (IoU) method and obtaining the False Positive (FP), False Negative (FN) and True Positive (TP) values. From these, the *F1 score* at the optimal threshold C_{thr1}, altogether with the mean Average Precision (mAP) values are calculated.

Precision represents the percentage of TP predictions with respect to all predictions (Equation (4)). *Recall* refers to the percentage of TP predictions with respect to all real instances present in the ground-truth data (Equation (5)).

$$Precision = \frac{TP}{TP + FP} , \tag{4}$$

$$Recall = \frac{TP}{TP + FN} . \tag{5}$$

Finally, the *F1 score* [13] is calculated for each sweep step from its corresponding *Precision* and *Recall* values (Equation (6)). The *F1 score* is a measure of overall accuracy. The C_{thr1} associated to the step with the highest *F1 score* is selected as the optimal C_{thr1}, obtaining the best *Precision* and *Recall* metrics.

$$F1score = 2 \times \frac{Recall \times Precision}{Recall + Precision} . \tag{6}$$

Since the main aim in our application is to detect and count the number of jellyfish, finding the optimal C_{thr1} is critical as we need a good trade-off between maximizing the prediction of jellyfish (TP) while minimizing the number of wrongly detected jellyfish (FP). The process that evaluated the prediction performance of the model is represented in Figure 6.

4.6. Real-Time Quantification

After training the network and selecting the best hyperparameter values and combination, we assessed the capability of the network at real-time jellyfish monitoring tasks. To do so, a video sequence was manually labeled, indicating the number and classes of jellyfish present at each frame. Subsequently, the same video was analyzed by the neural network. Each time that the network was able to analyze a frame for the video sequence, it generated a predicted information point, containing the number and classes of jellyfish present at the analyzed frame.

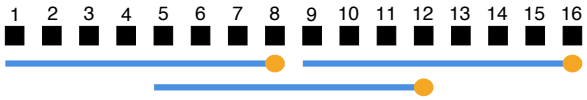


Figure 7: Representation of the window analysis and overlapping techniques ($W_{size} = 8$, $W_{over} = 50\%$). The black squares represent the predicted information points, the blue lines represent the windows and the orange dots are the resulting information point of each window.

The neural network detection may be affected by sporadic changes in luminosity, strange jellyfish positions, water reflexes, etc., resulting in the loss of TP detection or the appearance of FN detection. To minimize the effect of sporadic changes in the detection and improve the quantification performance of the neural network, we implemented diverse time series processing techniques.

Firstly, we performed a window analysis over the predicted information points. This technique takes the information of W_{size} number of predicted information points and processes it to generate a resulting information point (R_I_point). In our case, the value of the resulting information point was taken as the most occurring value from the analyzed window predicted information points. The application of this technique helps to eliminate sporadic detection errors. Three different window sizes were tested: $W_{size} = 4, 8, 12$ information points.

Secondly, we decided to apply an overlap between the information windows in order to preserve the significance of the predicted information points in the transition between windows. This overlap allows us to obtain resulting information points more frequently. Three different window overlaps were tested: $W_{over} = 25\%, 50\%, 75\%$. A representation of the application of these time series processing techniques over a series of predicted information points can be seen in Figure 7.

Due to the implementation of these techniques, the optimal confidence threshold to obtain the best *Similarity* is bound to diverge from the previously selected C_{thr1}. So, following the procedure explained in Section 4.5, we performed a threshold sweep over the confidence of the video sequence detection. For each threshold, we applied all combinations of windowing parameters. Finally, for each combination, the C_{thr2} that resulted in the best *Similarity* was selected.

The comparison between the manual and network predictions was carried out by computing the *Similarity* between the manual and neural network quantification, expressed as the percentage of correct resulting information points over the total number of resulting information points (Equation (7)). We classify an information point as correct when it correctly indicates the number and classes of jellyfish present in a determined time of the analyzed video.

$$Similarity = \frac{correct\ R\_I\_point}{total\ R\_I\_point} . \tag{7}$$

Table 2
Results obtained from the evaluation of each experiment *K*, indicating the hyperparameters used along with the AP obtained for each class, the mAP value, optimal C_{thr1} and corresponding *F1 score*.

Exp.	D. aug.	Lr.	Iter.	AP			mAP	C_{thr1}	F1 score
				P. noct.	R. pulmo	C. tuber.			
1	No	0.0005	10k	85.3%	98.2%	97.2%	93.6%	85%	93.7%
2			20k	86.3%	97.7%	97.3%	93.8%	85%	94.0%
3			40k	86.1%	97.6%	97.1%	93.6%	93%	94.1%
4		decay	10k	86.5%	98.1%	97.5%	94.0%	82%	94.1%
5			20k	86.3%	98.4%	97.3%	94.0%	95%	94.2%
6			40k	85.8%	98.9%	96.6%	93.8%	91%	94.2%
7	Yes	0.0005	10k	84.4%	97.5%	96.7%	92.9%	79%	93.6%
8			20k	86.5%	98.8%	96.7%	94.0%	91%	94.5%
9			40k	86.8%	**99.0%**	96.5%	94.1%	89%	94.8%
10		decay	10k	87.1%	98.5%	96.9%	94.1%	69%	94.6%
11			20k	87.6%	**99.0%**	97.5%	94.7%	86%	95.0%
12			40k	**88.2%**	**99.0%**	**97.7%**	**95.0%**	90%	**95.2%**

5. Results and Discussion

This section reports the performance obtained for each experiment in the final predictions and discusses the effect of each hyperparameter over it. Also, it exposes the real-time quantification results obtained from analyzing a video sequence and the conclusions that can be extracted from these. Finally, it presents a comparison between the performance of the selected Inception-ResNet V2 architecture versus two of its main competitors in both final predictions and quantification.

5.1. Experiment Performance

Average results obtained from the ten models corresponding to each one of the *K=1..12* experiments are shown in Table 2.

All experiments showed mAP values in the 93%–95% range, reaching a maximum of 95.0% for experiment 12 and a minimum value of 92.9% for experiment 7. The comparison of AP values for the three species shows that *R. pulmo* and *C. tuberculata* have higher AP values than *P. noctiluca*. This might be related to the fact that *R. pulmo* and *C. tuberculata* are bigger specimens and the shape of their bodies remains relatively unchanged while swimming and therefore they might be easier to identify. On the contrary, in *P. noctiluca* the relative position of the tentacles in relation to the main body (umbrella) changes to a greater extent with the movement of the animal, adopting a multitude of shapes, making it more difficult to identify. Regarding the C_{thr1} and *F1 score* values, most experiments found the best *F1 score* when applying relatively high C_{thr1} values, indicating that most TP detection had high confidence levels. Experiments showed *F1 scores* ranged from of 93% to 95%, reaching a maximum of 95.2% for experiment 12 again.

The comparison of the different experiments on a hyperparameter basis indicates that the application of data augmentation, the use of a higher number of iterations and the decay technique application resulted into increased performances. Experiment 12, which featured all three hyperparameters, presented the best performance. Figure 8 illustrates an example of the detection of jellyfish over images from the test set.

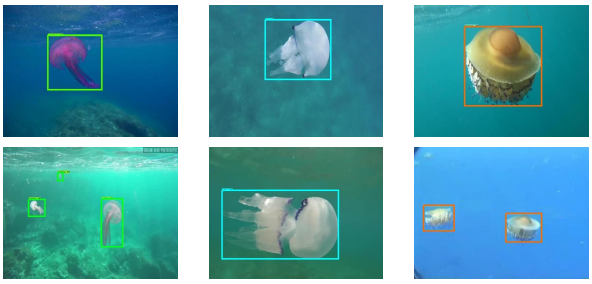


Figure 8: Jellyfish detection examples over test set images. Left: green bounding boxes over *P. noctiluca*; center: blue boxes over *R. pulmo*; right: orange bounding boxes over *C. tuberculata*.

5.2. Real-Time Quantification

To perform the quantification task and obtain its results, we made use of the best model *M* from experiment 12, containing the previously selected best-performing hyperparameters. We forwarded a 1920*x*1080 video sequence recorded by the authors using the procedures mentioned in Section 4.2 and analyzed it in real-time. No images from this video had been used either for training nor for testing the network. The duration of the video is approximately 5 minutes and contains a single jellyfish species (*P. noctiluca*) as, despite the best efforts, no videos with more than one of the studied jellyfish species could be located. This analysis was carried out in a computer with the following specs—processor: Intel i7-7700, RAM: 16 GB, GPU: NVIDIA GeForce GTX 1080).

Table 3 shows the obtained results for all windowing parameter combinations. The third column of the table indicates the time between resulting information points ($T_{R\_I\_point}$) in seconds after applying the time series processing techniques, obtained from Equation (8).

Table 3
Quantification results obtained from analyzing a video sequence for all windowing parameter combinations.

W_{size}	W_{over}	$T_{R\_I\_point}$	C_{thr2}	***Similarity***
4	25%	1.87	11%	87.7%
	50%	1.25	12%	87.9%
	75%	0.62	20%	87.5%
8	25%	3.75	36%	90.5%
	50%	2.50	36%	92.2%
	75%	1.25	36%	90.5%
12	25%	5.62	20%	**93.8%**
	50%	3.75	27%	92.7%
	75%	1.87	27%	92.1%

$$T_{R\_I\_point} = \frac{w\_size \times (1 - w\_overlap)}{fps}\ , \tag{8}$$

where fps indicates the frame rate at which the network was able to analyze the forwarded video. The Inception ResNet V2 architecture was able to perform the inference of a frame each 0.625 seconds (1.6 fps).

$T_{R\_I\_point}$ can be adjusted to meet the monitoring target requirements. The W_{size} could be lowered and the W_{over} raised to reduce this time, or the other way around to increase it.

It can be seen that all combinations showed high *Similarity* values, reaching a maximum of 93.8% when using a $W_{size} = 12$ predicted information points and an overlapping between windows of $W_{over} = 25\%$. Selecting these windowing parameters, a resulting information point is obtained each 5.62 seconds (following Equation (8)), endorsing that this value is adequate for the monitoring of slow-moving organisms such as jellyfish.

It can also be appreciated that the best *Similarity* for all combinations was achieved when applying much lower C_{thr2} than the C_{thr1} values obtained during the pure prediction task presented in Table 2. The time series processing techniques eliminate spurious FP predictions, allowing us to reduce the C_{thr2} values and introducing low confidence TP predictions while not being punished by low confidence FP.

The solidity of results using the quantification algorithm can be appreciated in Figure 9, which shows the difference between the jellyfish count obtained when using the final predictions versus the application of the quantification algorithm algorithm over the Inception ResNet V2 detection.

Figure 9a shows the count of each studied jellyfish species calculated from the final predictions. It can be seen how this value highly varies in time. Figure 9b shows the count obtained after the quantification algorithm using the windowing parameters that showed the best performance. The count is stable over time and closer to reality.

Additionally, the manually generated jellyfish count, acting as ground truth, for the same video is presented in Figure 10a along with its comparison against the obtained quantification in Figure 10b. The comparison has been made only for the *Pelagia noctiluca* species, as it was the only species present in the video sequence, thus, there is no quantification of errors for the other two species.

Table 4
Summary of detection performance metrics of Inception-ResNet V2, Inception V2 and ResNet101 neural network architectures.

Architecture	mAP	F1 score
Inception V2	76.5%	80.2%
ResNet	93.9%	94.2%
Incep.-ResNet V2	**95.2%**	**95.2%**

Figure 10 shows that some of the divergences can be found when the jellyfish count changes, where the network quantification shows a slower reaction compared to the manual quantification, caused by the computational time of the network and the $T_{R\_I\_point}$ introduced by the time series processing techniques. Also, some other quantification error are due to some timely close resulting information points containing detection errors.

An illustrative video of Jellytoring analyzing the studied video sequence can be seen on the SRV research group web page [47].

5.3. Neural Network Performance Comparison

To evaluate the effectiveness of the selected neural network and address its adequacy to our application in terms of detection performance and computational cost, we performed a comparison between the Inception-ResNet V2 architecture and two other object detection architectures, the InceptionV2 [65] and the ResNet101 [28].

These architectures were selected as they are close competitors to Inception-ResNet V2 in terms of detection performance and computational cost trade-off [23].

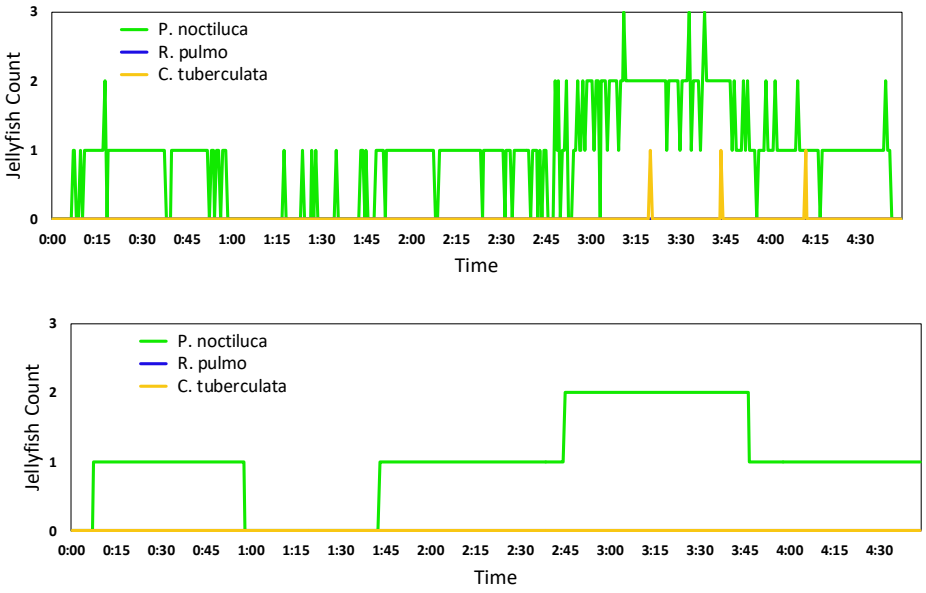
First, the three architectures were trained and tested over the dataset presented in Subsection 4.2 with the selected best hyperparameters from Subsection 5.1 and the 10k-fold cross-validation strategy. The detection performance comparison was conducted using the mAP and *F1 score* evaluation metrics. Table 4 shows the comparison between detection performance metrics.

The mAp and *F1 score* comparison among the three architectures indicates that Inception-ResNet V2 offers the highest detection performance. ResNet101 architecture shows detection metrics close to those of Inception-ResNet V2 albeit slightly lower. Conversely, Inception V2 shows worse mAP and *F1 score* values.

Following, the video sequence presented in Section 5.2 was forwarded into the three architectures and their detection's were processed by the quantification algorithm.

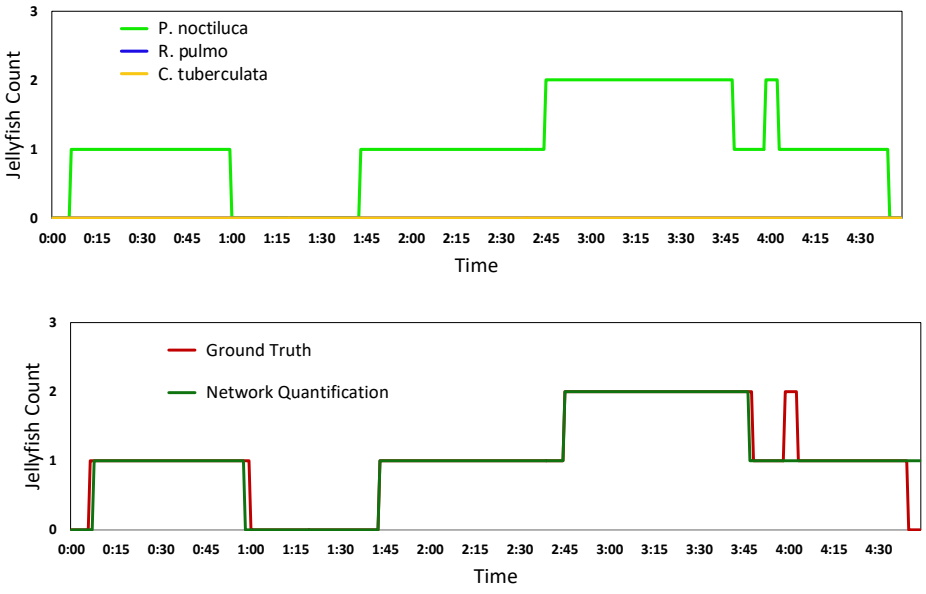
Table 5 exposes the comparison between quantification results. The presented *Similarity* results are from the best C_{thr2} for each combination. The W_{size} values were adjusted, taking into account each network fps, to maintain the same time between resulting information points as the ones obtained in Table 3

In terms of fps achieved, the Inception V2 architecture was able to analyze 25.2 frames per second, while the



(a) (b)

Figure 9: Results of the jellyfish count from Inception ResNet V2 final predictions (a) and quantification algorithm (b) over a video showcasing nearly 5 minutes of footage of up to two *P. noctiluca* jellyfish going in and out of the frame.



(a) (b)

Figure 10: Results of the jellyfish count from manually generated ground truth (a) and its comparison against the results from the Inception ResNet V2 network quantification algorithm (b).

ResNet101 managed to process 10 frames per second. Both architectures achieve higher fps values than the Inception-ResNet V2 architecture (1.6), meaning that higher W_{size} values can be used to incorporate more predicted information points in each window, helping to reduce spurious detection errors. Nevertheless, it can be seen that neither the Inception V2 nor the ResNet101 architectures were able to obtain higher *Similarity* values than the Inception ResNet V2, reaching 73.3% and 90.0%, respectively.

Figures 11a and 12a show the results of the jellyfish count from the ResNet101 and Inception V2 network final predictions, respectively. In the same way, Figure 11b and 12b present the corresponding network quantification when using the best windowing parameters.

Table 5
Quantification results of Inception-ResNet V2, Inception V2 and ResNet101 neural network architectures.

Inception-V2 fps achieved: 25.2

W_{size}	W_{over}	*Similarity*
63	25%	70.3%
	50%	70.7%
	75%	72.1%
126	25%	70.0%
	50%	72.0%
	75%	71.3%
189	25%	69.9%
	50%	**73.3%**
	75%	70.9%

ResNet101 fps achieved: 10.0

W_{size}	W_{over}	*Similarity*
25	25%	**90.0%**
	50%	89.3%
	75%	89.0%
50	25%	87.8%
	50%	87.5%
	75%	86.7%
75	25%	87.8%
	50%	86.3%
	75%	84.8%

Inception-ResNet V2 fps achieved: 1.6

W_{size}	W_{over}	*Similarity*
4	25%	87.7%
	50%	87.9%
	75%	87.5%
8	25%	90.5%
	50%	92.2%
	75%	90.5%
12	25%	**93.8%**
	50%	92.7%
	75%	92.1%

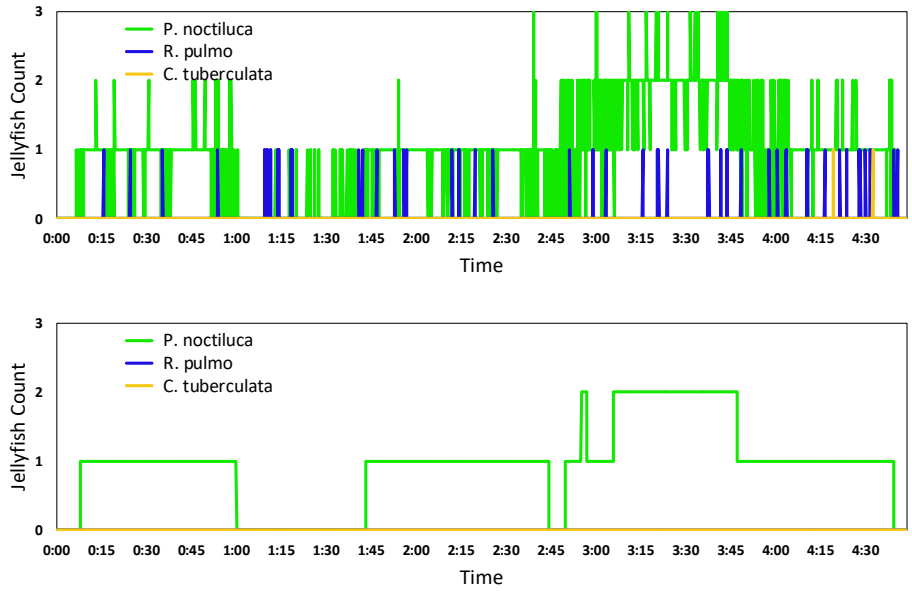


Figure 11: Results of the jellyfish count from the ResNet101 network final predictions (a) and quantification algorithm (b).

The high detection and quantification metrics shown by the Inception-ResNet V2 network make it the most suitable for jellyfish monitoring. The ResNet101 architecture offers a moderate trade-off between computational cost and quantification performance, still reaching good detection and quantification metrics at higher frames per second, making it suitable for detecting and quantifying faster species. The Inception V2 architecture offers a more extreme trade-off between computation cost and quantification performance, providing much lower inference time at still reasonably good detection and quantification metrics.

6. Conclusions and Future Work

This paper presents Jellytoring, a system for real-time jellyfish monitoring from underwater video recordings. Jellytoring uses a deep object detection neural network to detect and classify jellyfish instances, combined with a quantification algorithm. A main advantage of this system is that it is able to automatically monitor jellyfish presence without the need for any human interaction, allowing us to generate continuous and precise records. Additionally, the information can be fed to the system in real-time, generating live records.

The neural network evaluation presented very high metrics in the prediction task, reaching a maximum *F1 score* of 95.2% when the data augmentation and learning rate decay techniques were applied and the network was trained for 40,000 iterations. On the same page, the best quantification results were obtained when choosing a W_{size} of 12 information points and a W_{over} of 25%, being able to analyze a video sequence with a *Similarity* of 93.8% between the manually generated ground truth and the output of the quantification algorithm. These results indicate that the presented system is able to detect, quantify and monitor jellyfish with high accuracy, thanks to the quantification algorithm that improves the neural network detection.

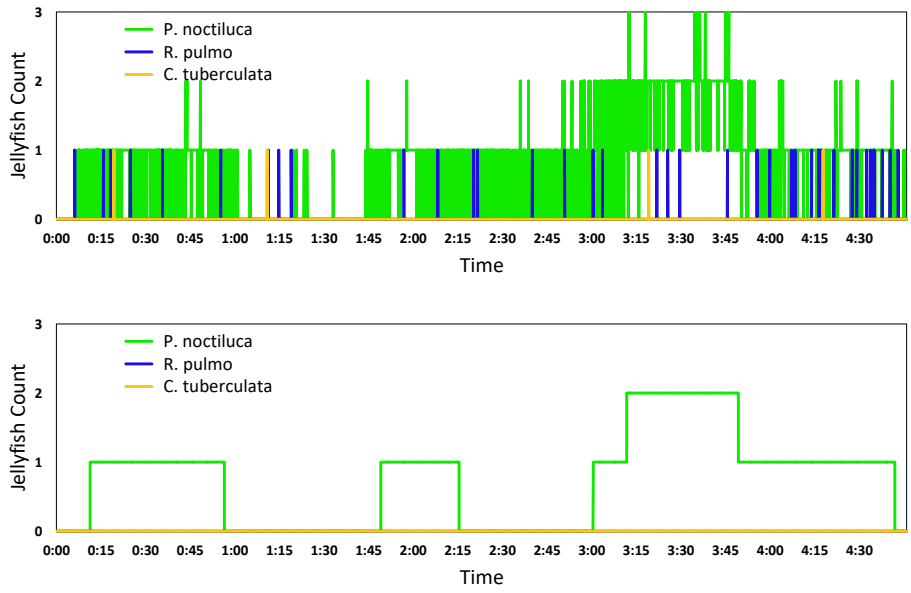


Figure 12: Results of the jellyfish count from the Inception V2 network final predictions (a) and quantification algorithm (b).

Additionally, Jellytoring can be customized, widening the applicability of the system. This can be done either by using other network architectures or changing the windowing parameters from the time series processing techniques. Some other possible applications could be the monitoring of other jellyfish species, faster species such as fish, or even other objects like marine waste.

Further developments will focus on lightening the system computational load while maintaining high accuracy levels. Also, we will work on increasing the number of jellyfish species the network can distinguish, widening its spatial application. Our final goal is to implement this system on a floating station and be executed online to monitor the presence and class of jellyfish and relate it to determined water conditions.

We provide our dataset and code, along with the best trained inference frozen model in a GitHub repository [46].

Acknowledgments

Miguel Martin-Abadal was supported by Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contract DPI2017-86372-C3-3-R. Ana Ruiz-Frau was supported by a Marie-Sklodowska-Curie Individual Fellowship (JellyPacts project number 655475). Hilmar Hinz was supported through a Ramón y Cajal Fellowship financed by the Ministerio de Economía y Competitividad de España and the Conselleria d'Educació, Cultura i Universitats Comunidad Autónoma de las Islas Baleares (RyC 2013 14729). Yolanda Gonzalez-Cid was supported by Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contracts TIN2017-85572-P and DPI2017-86372-C3-1-R.

The authors would like to thank Charlotte Jennings for her help in the collection of data and analysis of the images.

CRediT authorship contribution statement

Miguel Martin-Abadal: Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing. **Ana Ruiz-Frau:** Conceptualization, Methodology, Investigation, Data Curation, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Hilmar Hinz:** Conceptualization, Methodology, Investigation, Data Curation, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Yolanda Gonzalez-Cid:** Investigation, Methodology, Validation, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

[1] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B., 2018. Sanity checks for saliency maps. Advances in Neural Information Processing Systems 31 , 9505–9515URL: http://papers.nips.cc/paper/8160-sanity-checks-for-saliency-maps.pdf.

[2] Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. Neural Networks: Tricks of the Trade .

[3] Bodla, N., Singh, B., Chellappa, R., Davis, L.S., 2017. Soft-nms — improving object detection with one line of code. 2017 IEEE International Conference on Computer Vision (ICCV) , 5562–5570.

[4] Borowicz, A., McDowall, P., Youngflesh, C., Sayre-McCord, T., Clucas, G., Herman, R., Forrest, S., Rider, M., Schwaller, M., Hart, T., Jenouvrier, S., Polito, M.J., Singh, H., Lynch, H.J., 2018. Multimodal survey of Adélie penguin mega-colonies reveals the Danger Islands as a seabird hotspot. Scientific Reports 8, 3926. URL: http://www.nature.com/articles/s41598-018-22313-w.

[5] Buil, M.D., 2011. NON-MAXIMA SUPRESSION. Technical Report. Computer Graphics and Vision, Graz University of Technology, Austria.

[6] Caughlan, L., 2001. Cost considerations for long-term ecological monitoring. Ecological Indicators 1, 123–134. URL: http://linkinghub.elsevier.com/retrieve/pii/S1470160X01000152.

[7] Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20, 273–297. URL: https://doi.org/10.1007/BF00994018.

[8] Dai, J., Li, Y., He, K., Sun, J., 2016. R-fcn: Object detection via region-based fully convolutional networks. NIPS .

[9] Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2.

[10] Del Vecchio, S., Fantinato, E., Silan, G., Buffa, G., 2019. Trade-offs between sampling effort and data quality in habitat monitoring. Biodiversity and Conservation 28, 55–73. URL: http://link.springer.com/10.1007/s10531-018-1636-5.

[11] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: A Large-Scale Hierarchical Image Database. 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) .

[12] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The PASCAL Visual Object Classes (VOC) challenge. International Journal of Computer Vision 88, 303–338.

[13] F1 score, 2018. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/F1_score. [Online; accessed 23-March-2019].

[14] Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D., 2009. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1627–1645.

[15] Fenner, P.J., Lippmann, J., Gershwin, L., 2010. Fatal and Nonfatal Severe Jellyfish Stings in Thai Waters. Journal of Travel Medicine 17, 133–138. URL: https://academic.oup.com/jtm/article-lookup/doi/10.1111/j.1708-8305.2009.00390.x.

[16] French, G., Mackiewicz, M., Fisher, M., Challis, M., Knight, P., Robinson, B., Bloomfield, A., 2018. Jellymonitor: automated detection of jellyfish in sonar images using neural networks. Proceedings of the 14th IEEE International Conference on Signal Processing .

[17] Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119 – 139. URL: http://www.sciencedirect.com/science/article/pii/S002200009791504X.

[18] Galil, B.S., 2007. Loss or gain? Invasive aliens and biodiversity in the Mediterranean Sea. Marine Pollution Bulletin 55, 314–322.

[19] Geisser, S., 1975. The predictive sample reuse method with applications. Journal of the American Statistical Association 70, 320–328. doi:doi: 10.1080/01621459.1975.10479865.

[20] Girshick, R., 2015. Fast r-cnn. 2015 IEEE International Conference on Computer Vision (ICCV) , 1440–1448.

[21] Girshick, R.B., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 580–587.

[22] Gomez Villa, A., Salazar, A., Vargas, F., 2017. Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. Ecological Informatics 41, 24–32. URL: https://www.sciencedirect.com/science/article/pii/S1574954116302047.

[23] Google-Tensorflow, 2018. Coco-trained models. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md.

[24] Graham, W.M., Martin, D.L., Martin, J.C., 2003. In situ quantification and analysis of large jellyfish using a novel video profiler. Marine Ecology Progress Series 254, 129–140.

[25] Gray, P.C., Fleishman, A.B., Klein, D.J., McKown, M.W., Bézy, V.S., Lohmann, K.J., Johnston, D.W., 2018. A Convolutional Neural Network for Detecting Sea Turtles in Drone Imagery. Methods in Ecology and Evolution In Review, 1–11.

[26] Halpern, B., Walbridge, S., Selkoe, K., Kappel, C., Micheli, F., D'Agrosa, C., Bruno, J., Casey, K., Ebert, C., Fox, H., Fujita, R., Heinemann, D., S Lenihan, H., M P Madin, E., T Perry, M., Selig, E., Spalding, M., Steneck, R., Watson, R., 2008. A global map of human impact on marine ecosystems. Science (New York, N.Y.) 319, 948–52.

[27] He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. arXiv:1703.06870.

[28] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

[29] Holmes, T.H., Wilson, S.K., Travers, M.J., Langlois, T.J., Evans, R.D., Moore, G.I., Douglas, R.A., Shedrawi, G., Harvey, E.S., Hickey, K., 2013. A comparison of visual- and stereo-video based fish community assessment methods in tropical and temperate marine waters of Western Australia. Limnology and Oceanography: Methods 11, 337–350.

[30] Hong, S.J., Han, Y., Kim, S.Y., Lee, A.Y., Kim, G., 2019. Application of Deep-Learning Methods to Bird Detection Using Unmanned Aerial Vehicle Imagery. Sensors 19, 1651. URL: https://www.mdpi.com/1424-8220/19/7/1651.

[31] Houghton, J., Doyle, T., Davenport, J., Hays, G., 2006. Developing a simple, rapid method for identifying and monitoring jellyfish aggregations from the air. Marine Ecology Progress Series 314, 159–170.

[32] Hughes, T.P., Baird, A.H., Bellwood, D.R., Card, M., Connolly, S.R., Folke, C., Grosberg, R., Hoegh-Guldberg, O., Jackson, J.B., Kleypas, J., Lough, J.M., Marshall, P., Nyström, M., Palumbi, S.R., Pandolfi, J.M., Rosen, B., Roughgarden, J., 2003. Climate change, human impacts, and the resilience of coral reefs. Science 301, 929–933. URL: https://www.jstor.org/stable/3834832.

[33] Islam, M.S., Tanaka, M., 2004. Impacts of pollution on coastal and marine ecosystems including coastal and marine fisheries and approach for management: A review and synthesis. Marine Pollution Bulletin .

[34] Kaiser, M., Collie, J., Hall, S., Jennings, S., Poiner, I., 2002a. Modification of marine habitats by trawling activities: Prognosis and solutions. Fish and Fisheries 3, 114 – 136.

[35] Kaiser, M., Collie, J., Hall, S., Jennings, S., Poiner, I., 2002b. Modification of marine habitats by trawling activities: Prognosis and solutions. Fish and Fisheries 3, 114 – 136.

[36] Kim, H., Koo, J., Kim, D., Jung, S., Shin, J., Lee, S., Myung, H., 2016. Image-based monitoring of jellyfish using deep learning architecture. IEEE Sensors Journal 16, 2215–2216.

[37] Langlois, T.J., Harvey, E.S., Fitzpatrick, B., Meeuwig, J.J., Shedrawi, G., Watson, D.L., 2010. Cost-efficient sampling of fish assemblages: Comparison of baited video stations and diver video transects. Aquatic Biology 9, 155–168.

[38] Lee, J., HW, C., Chae, J., Kim, D., Lee, S., 2006. Performance analysis of intake screens in power plants on mass impingement of marine organisms. Ocean and polar research 28, 385–393.

[39] Levy, D., Levy, D., Belfer, Y., Osherov, E., Bigal, E., Scheinin, A.P., Nativ, H., Tchernov, D., Treibitz, T., 2018. Automated analysis of marine video with limited data. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) , 1466–14668doi:doi: 10.1109/CVPRW.2018.00187.

[40] Li, X., Shang, M., Hao, J., Yang, Z., 2016. Accelerating fish detection and recognition by sharing cnns with objectness learning. OCEANS 2016 - Shanghai , 1–5doi:doi: 10.1109/OCEANSAP.2016.7485476.

[41] Lienhart, R., Maydt, J., 2002. An extended set of haar-like features for rapid object detection. Proceedings of the International Conference on Image Processing 1, I–900.

[42] Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 936–944doi:doi: 10.1109/CVPR.2017.106.

[43] Lin, T.Y., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. European Conference

on Computer Vision (ECCV) .

[44] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. Computer Vision – ECCV 2016 , 21–37.

[45] Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–.

[46] Martin-Abadal, M., 2019. Jellyfish object detection. https://github.com/srv/jf_object_detection.

[47] Martin-Abadal, M., Ruiz-Frau, A., Gonzalez-Cid, Y., 2019. Video: Real-time jellyfish detection and quantification. http://srv.uib.es/jellyfish-quantification/.

[48] Matsumura, K., Kamiya, K., Yamashita, K., Hayashi, F., Watanabe, I., Murao, Y., Miyasaka, H., Kamimura, N., Nogami, M., 2005. Genetic polymorphism of the adult medusae invading an electric power station and wild polyps of Aurelia aurita in Wakasa Bay, Japan. Journal of the Marine Biological Association of the UK 85, 563–568. URL: http://www.journals.cambridge.org/abstract{_}S0025315405011483.

[49] Merceron, M., Le Fevre-Lehoerff, G., Bizouarn, Y., Kempf, M., 1995. Fish and jellyfish in Brittany (France). Equinoxe 56, 6–8.

[50] Millennium Ecosystem Assessment, 2005. Ecosystems and human well-being: a framework working group for assessment report of the Millennium Ecosystem Assessment. Island Press, Washington.

[51] Moniruzzaman, M., Islam, S.M.S., Bennamoun, M., Lavery, P., 2017. Deep learning on underwater marine object detection: A survey. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10617 LNCS, 150–160. doi:doi: 10.1007/978-3-319-70353-4_13.

[52] Neubeck, A., Van Gool, L., 2006. Efficient non-maximum suppression. Proceedings of International Conference on Pattern Recognition 3, 850–855.

[53] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J., 2020. Deep learning vs. traditional computer vision. Advances in Computer Vision , 128–144URL: https://app.dimensions.ai/details/publication/pub.1113641911, doi:doi: 10.1007/978-3-030-17795-9_10.

[54] Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28 , III–1310–III–1318URL: http://dl.acm.org/citation.cfm?id=3042817.3043083.

[55] Pauly, D., Watson, R., Alder, J., 2005. Global trends in world fisheries: Impacts on marine ecosystems and food security. Philosophical Transactions of the Royal Society B: Biological Sciences 360, 5–12.

[56] Pierce, J., 2009. Prediction, location, collection and transport of jellyfish (cnidaria) and their polyps. Zoo biology 28, 163–76.

[57] Purcell, J.E., Baxter, E.J., Fuentes, V.L., 2013. Jellyfish as products and problems of aquaculture. Advances in Aquaculture Hatchery Technology , 404–430URL: https://linkinghub.elsevier.com/retrieve/pii/B9780857091192500139.

[58] Purcell, J.E., Uye, S.i.I., Lo, W.T.T., 2007. Anthropogenic causes of jellyfish blooms and their direct consequences for humans: a review. Marine Ecology Progress Series 350, 153–174. URL: http://www.int-res.com/articles/meps2007/350/m350p153.pdf.

[59] Py, O., Hong, H., Zhongzhi, S., 2016. Plankton classification with deep convolutional neural networks. 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference , 132–136URL: http://ieeexplore.ieee.org/document/7560334/.

[60] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 779–788doi:doi: 10.1109/CVPR.2016.91.

[61] Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39.

[62] Rife, J., Rock, S.M., 2003. Segmentation methods for visual tracking of deep-ocean jellyfish using a conventional camera. IEEE Journal of Oceanic Engineering 28, 595–608.

[63] Szegedy, C., Ioffe, S., Vanhoucke, V., 2016a. Inception-v4, inception-resnet and the impact of residual connections on learning. AAAI Conference on Artificial Intelligence .

[64] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 00, 1–9. URL: doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298594.

[65] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016b. Rethinking the inception architecture for computer vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 2818–2826.

[66] Taylor, L., Nitschke, G., 2017. Improving Deep Learning using Generic Data Augmentation. ArXiv e-prints - 1708.06020 arXiv:1708.06020.

[67] Tzutalin, D., 2018. Labelimg. https://github.com/tzutalin/labelImg.

[68] Valletta, J.J., Torney, C., Kings, M., Thornton, A., Madden, J., 2017. Applications of machine learning in animal behaviour studies. Animal Behaviour 124, 203–220. URL: https://www.sciencedirect.com/science/article/pii/S0003347216303360.

[69] Villon, S., Chaumont, M., Subsol, G., Villéger, S., Claverie, T., Mouillot, D., 2016. Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between deep learning and hog+svm methods. Advanced Concepts for Intelligent Vision Systems , 160–171.

[70] Wäldchen, J., Mäder, P., 2018. Machine learning for image based species identification. Methods in Ecology and Evolution 9, 2216–2225.

[71] Willi, M., Pitman, R.T., Cardoso, A.W., Locke, C., Swanson, A., Boyer, A., Veldthuis, M., Fortson, L., 2018. Identifying animal species in camera trap images using deep learning and citizen science. Methods in Ecology and Evolution 2018, 1–12.

[72] Xiu Li, Min Shang, Qin, H., Liansheng Chen, 2015. Fast accurate fish detection and recognition of underwater images with fast r-cnn. OCEANS 2015 - MTS/IEEE Washington , 1–5.

[73] Zhu, M., 2004. Recall, precision and average precision. Technical Report. Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Canada.

Chapter 5

Conclusions

This chapter summarises the contributions of this thesis and analyses the research relevance, main findings, and drawn conclusions. Finally, it presents some areas of improvement and possible future lines of research.

5.1 Contributions and discussion

The main objective of this thesis was to develop deep learning-based tools using CNNs for image and video processing and to implement them in real-world scenarios for marine ecosystem services preservation tasks. It also aimed to design, test, and validate a methodology for the development and efficient implementation of these tools.

This thesis presents three different tools, each tackling a specific task with varying requirements. Diverse types of deep CNNs were used, and their applicability was tested across a wide range of scenarios.

Following, the main objectives and contributions for each task are presented. Specific scenarios where CNNs have been implemented are also detailed, discussing the selected CNNs architecture types, data gathering methods, and deployment platforms.

1. ***Posidonia oceanica* monitoring**

The objective was to develop a tool to automatically perform high-precision semantic segmentation of *Posidonia oceanica* meadows and their habitat in sea-floor images using deep learning techniques. The following work was carried out:

- Dataset generation: 483 images containing *Posidonia oceanica* meadows and their habitat were gathered from six immersions conducted on different Mediterranean sea locations at depths ranging from 2-20 meters. The images were taken using multiple cameras mounted on an AUV and under diverse environmental conditions such as sunlight or water turbidity, ensuring robust network training. Additionally, semantic segmentation ground truths were generated.
- CNN implementation: Considering that *Posidonia oceanica* grows in dense meadows of irregular shapes and, equally, sea-floor substrates do not have a defined shape, CNN semantic segmentation architectures were selected as the most adequate approach. These architectures are able to perform pixel-wise classification, distinguishing multiple areas in an image without shape restrictions. The selected network was the VGG16-FCN8 (Simonyan and Zisserman, 2014) and, after selecting the best-performing hyperparameters, it achieved AUC values of 97.7% when performing a binary classification between *Posidonia oceanica* and background, and of 96.8% when distinguishing between *Posidonia oceanica*, rock and sand substratum.
- Deployment: The output layer of the CNN was adapted to reduce the inference time, allowing online execution. Additionally, integration into AUV and ASV platforms was performed using the *ROS* middleware.

This work was developed under the "DEvelopment of new TEChnologies for the automatic and periodic assessment of changes in POSidonia meadows due to anthropogenic causes" (DETECPOS) project (SRV, 2020) and has been used to generate offline *Posidonia oceanica* semantic maps of large areas for its control and monitoring (Gonzalez-Cid et al., 2021). Additionally, it has been deployed in an AUV, performing online image segmentation, serving as an input source to a generation of online semantic coverage maps (Guerrero-Font et al., 2021b) and to a decision-time adaptive replanning algorithm to dynamically adapt the robot exploration using the visual information gathered online (Guerrero, Bonin-Font, and Oliver, 2021).

The generated dataset, trained models, and additional code are provided to the scientific community in (**Martin-Abadal, Miguel**, 2018).

2. **Pipeline Characterisation**

The objective was to design a system able to automatically identify and characterise valves, pipes, and structural elements on underwater pipeline networks and position them in a 3D space to provide information during inspection and manipulation tasks. The following work was carried out:

- Dataset generation: 606 point clouds showcasing a wide variety of pipe structures and valve connections over different backgrounds were gathered. The point clouds were generated from pairs of images provided by different stereo camera rigs mounted on an AUV and an ASV. The images were taken under diverse environmental conditions to ensure robust training. Additionally, 3D semantic segmentation ground truths were generated.
- CNN implementation: Underwater pipeline structures range from simpler ones, like pipelines laid on the seabed covering large distances, to more complex ones, such as the pipe and valve layouts found in oil rigs. In all cases, it is important to analyse and extract 3D information from unknown-shaped objects and calculate sizes, gripping points, lengths, etc. Thus, 3D CNN semantic segmentation architectures were selected as the most adequate approach. These architectures are able to perform pixel-wise classification, distinguishing multiple areas in a point cloud without shape restrictions. The selected network was the Dynamic Graph Convolutional Neural Network (*DGCNN*) (Wang et al., 2019) and, after selecting the best-performing hyperparameters, it reached a pixel-wise segmentation F1-score of 87.2%.
- Information processing: Generation of an information extraction algorithm that clusters the pixel-wise information to an instance level, raising the instance-level segmentation F1-score to 95.4%. This algorithm also draws information from the detected pipes and valves, providing lengths, centre and gripping points, and detecting pipe elbows and connections, with very little positioning error.
- Information processing: Generation of an information unification algorithm that merges the information of diverse point clouds provided by the information extraction algorithm and generates information maps of an inspected area.
- Deployment: Adapt the neural network and information algorithms for online execution and integration into AUV and ASV platforms using *ROS* middleware, for surveying and manipulation tasks.

This work was framed on the "TWIN roBOTs for cooperative underwater intervention missions" (TWINBOT) project (SRV, 2018), which aimed to achieve a step forward beyond the current underwater intervention state of the art and the development of a new kind of I-AUVs, able to work autonomously, alone or in a cooperative way. Currently, the "COOPErative Resident robots for Autonomous ManipulatiOn Subsea" (COOPERAMOS) project (SRV, 2021) has taken its place and aims to use at least three I-AUVs, cooperating to enable complex underwater intervention tasks, such as bulky load transport and cooperative complex structure assembly, in a priori unknown area, including obstacles, with high autonomy. The generated dataset, trained models and additional code are provided to the scientific community in (**Martin-Abadal, Miguel** et al., 2021a; **Martin-Abadal, Miguel**, Oliver-Codina, and Gonzalez-Cid, 2022a).

3. **Jellyfish detection and quantification**

The objective was to develop a tool able to automatically detect and quantify different species of jellyfish and log their presence during long periods of time. The following work was carried out:

- Dataset generation: 842 images containing instances of three different species of jellyfish were gathered. The images were extracted from publicly available videos on diverse social media sites. Additionally, object detection ground truths were generated.
- CNN implementation: Monitoring jellyfish populations and trends requires an effective system capable of identifying the number and species of jellyfish present in an area, enabling temporal quantification. To do so, CNN object detection architectures were selected as the most suitable approach. These architectures can localise and classify different object instances in an image. The selected network was the Inception ResNet v2 (Szegedy, Ioffe, and Vanhoucke, 2016) and, after selecting the best-performing hyperparameters, it reached an F1-score of 95.2% in the jellyfish detection task.

– Information processing: Generation of a quantification algorithm based on windowing techniques to log the presence of jellyfish over video sequences.

– Deployment: Adapt the neural network and quantification algorithm for an online execution, ready for integration into stationary marine buoys equipped with cameras.

This work has generated great interest among biologists. A second implementation of this tool has been developed (Ruiz-Frau et al., 2022), including a larger number of jellyfish species and a division between different oceanic regions, with specifically trained models, considering determined jellyfish species. Furthermore, a web page that will allow uploading images for online jellyfish detection and quantification, while providing extra data to enrich the dataset, is under development (Bustos, 2022). The generated dataset, trained models, and additional code are provided to the scientific community in (**Martin-Abadal, Miguel**, 2020).

These implementations cover a wide spectrum of scenarios where deep CNN have been applied with good results, obtaining high accuracy metrics and even surpassing humans in certain applications. They automate the data analysis process, allowing for temporal and spatial extension of the scope of analysis or surveys, and improve the repeatability of experiments to detect evolution trends. Additionally, all implementations have been, or are ready to be, deployed and executed in real-time on diverse platforms. Finally, they have proved their usefulness, as biologists have used them to obtain information during exploration campaigns, and have been integrated into other scientific works as a source of information. Thus, validating the methodology presented in Section 1.2 and proving the feasibility of implementing deep CNNs in challenging environments like marine environments, where data is often scarce and affected by light transmission artefacts or other environmental factors.

5.2 Future Work

Besides the specific future research lines identified for each presented tool, which are described in the "Conclusion" or "Future Work" sections of their corresponding publications, this thesis has identified several potential lines of future work and points for improvement in the design and implementation of deep learning tools for environmental applications.

- Improve data storage and accessibility with enriched metadata and ground truth annotations. Deep learning architectures need to be trained with lots of data, which sometimes can be scarce or inaccessible. It is important that the community moves towards open-source approaches, facilitating progress in the field.
- Study techniques to increase contact between biologists or environment experts and developers. It is crucial that both parties provide continuous feedback in order to assure a good understanding of the problem and the required system characteristics and features.
- Explore the implementation of semi-supervised or unsupervised deep learning approaches. Data curation and ground truth generation can be a time-consuming and tedious task due to the high volume of required data. These approaches could improve the obtained results and ease the workload, focusing the research on the exploration of new applications or solutions.
- Study the implementation of 3D information in deep learning environmental applications. During the work carried out for pipeline characterisation, the usefulness of working with 3D information was featured. Most CNN applications in the fields of biology and conservation use 2D information, albeit the many benefits 3D information can provide. In object detection and classification, 3D information could be used to identify new features on the studied species or objects, to size them, or to detect their pose. On broader analysis, using semantic segmentation, like seafloor inspection and identification, 3D information could provide the dimensions of a covered area, or even allow to calculate the volume of areas of interest, such as seagrass meadows.

Bibliography

Ahmad, Sajjad, Zahoor Ahmad, Cheol-Hong Kim, and Jong-Myon Kim (2022). "A Method for Pipeline Leak Detection Based on Acoustic Imaging and Deep Learning". In: *Sensors* 22.4. ISSN: 1424-8220. DOI: 10.3390/s22041562. URL: https://www.mdpi.com/1424-8220/22/4/1562.

Alonso, Iñigo, Matan Yuval, Gal Eyal, Tali Treibitz, and Ana C. Murillo (2019). "CoralSeg: Learning coral segmentation from sparse annotations". In: *Journal of Field Robotics* 36.8, pp. 1456–1477. DOI: https://doi.org/10.1002/rob.21915. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21915. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21915.

Ani, Chinenye J. and Barbara Robson (2021). "Responses of marine ecosystems to climate change impacts and their treatment in biogeochemical ecosystem models". In: *Marine Pollution Bulletin* 166, p. 112223. ISSN: 0025-326X. DOI: https://doi.org/10.1016/j.marpolbul.2021.112223. URL: https://www.sciencedirect.com/science/article/pii/S0025326X21002575.

Antao, Laura, Amanda Bates, Shane Blowes, Conor Waldock, Sarah Supp, Anne Magurran, Maria Dornelas, and Aafke Schipper (July 2020). "Temperature-related biodiversity change across temperate marine and terrestrial systems". In: *Nature Ecology & Evolution* 4, 927–933. DOI: 10.1038/s41559-020-1185-7.

Bacheler, Nathan M., Nathan R. Geraldi, Michael Ladd Burton, Roldan C Muñoz, and G. Todd Kellison (2017). "Comparing relative abundance, lengths, and habitat of temperate reef fishes using simultaneous underwater visual census, video, and trap sampling". In: *Marine Ecology Progress Series* 574, pp. 141–155.

Barbier, Edward B. (2017). "Marine ecosystem services". In: *Current Biology* 27.11, R507–R510. ISSN: 0960-9822. DOI: https://doi.org/10.1016/j.cub.2017.03.020. URL: https://www.sciencedirect.com/science/article/pii/S0960982217302890.

Bennett, Elena M., Garry D. Peterson, and Line J. Gordon (2009). "Understanding relationships among multiple ecosystem services". In: *Ecology Letters* 12.12, pp. 1394–1404. DOI: https://doi.org/10.1111/j.1461-0248.2009.01387.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1461-0248.2009.01387.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1461-0248.2009.01387.x.

Bermant, Peter, Michael Bronstein, Robert Wood, Shane Gero, and David Gruber (Aug. 2019). "Deep Machine Learning Techniques for the Detection and Classification of Sperm Whale Bioacoustics". In: *Scientific Reports* 9, pp. 1–10. DOI: 10.1038/s41598-019-48909-4.

Bharti, Vibhav, David Lane, and Sen Wang (2020). "Learning to Detect Subsea Pipelines with Deep Segmentation Network and Self-Supervision". In: *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pp. 1–7. DOI: 10.1109/IEEECONF38699.2020.9389226.

Borja, Angel (2014). "Grand challenges in marine ecosystems ecology". In: *Frontiers in Marine Science* 1. ISSN: 2296-7745. DOI: 10.3389/fmars.2014.00001. URL: https://www.frontiersin.org/articles/10.3389/fmars.2014.00001.

Brotz, Lucas, William W L Cheung, Kristin Kleisner, Evgeny Pakhomov, and Daniel Pauly (2012). "Increasing jellyfish populations: trends in Large Marine Ecosystems". In: *Hydrobiologia* 690 (1). PT: J; TC: 9, pp. 3–20. DOI: 10.1007/s10750-012-1039-7.

Buonocore, Elvira, Luigia Donnarumma, Luca Appolloni, Antonino Miccio, Giovanni F. Russo, and Pier Paolo Franzese (2020). "Marine natural capital and ecosystem services: An environmental accounting model". In: *Ecological Modelling* 424, p. 109029. ISSN: 0304-3800. DOI: https://doi.org/10.1016/j.ecolmodel.2020.109029. URL: https://www.sciencedirect.com/science/article/pii/S0304380020301010.

Burguera, Antoni (2020). "Segmentation through patch classification: A neural network approach to detect Posidonia oceanica in underwater images". In: *Ecological Informatics* 56, p. 101053. ISSN: 1574-9541. DOI: https://doi.org/10.1016/j.ecoinf.2020.101053. URL: https://www.sciencedirect.com/science/article/pii/S1574954120300030.

Burguera, Antoni and Francisco Bonin-Font (2020). "On-Line Multi-Class Segmentation of Side-Scan Sonar Imagery Using an Autonomous Underwater Vehicle". In: *Journal of Marine Science and Engineering* 8.8. ISSN: 2077-1312. DOI: 10.3390/jmse8080557. URL: https://www.mdpi.com/2077-1312/8/8/557.

Bustos, Rubén (2022). *Jellyfish Object Detection*. https://jellytoring.uib.es/.
Caughlan, L (2001). "Cost considerations for long-term ecological monitoring". In: *Ecological Indicators* 1.2, pp. 123–134.
Chai, Junyi, Hao Zeng, Anming Li, and Eric W.T. Ngai (2021). "Deep learning in computer vision: A critical review of emerging techniques and application scenarios". In: *Machine Learning with Applications* 6, p. 100134. ISSN: 2666-8270. DOI: https://doi.org/10.1016/j.mlwa.2021.100134. URL: https://www.sciencedirect.com/science/article/pii/S2666827021000670.
Character, Leila, Agustin Ortiz JR, Tim Beach, and Sheryl Luzzadder-Beach (2021). "Archaeologic Machine Learning for Shipwreck Detection Using Lidar and Sonar". In: *Remote Sensing* 13.9. ISSN: 2072-4292. DOI: 10.3390/rs13091759. URL: https://www.mdpi.com/2072-4292/13/9/1759.
Chen, Fu-Chen and Mohammad R. Jahanshahi (2018). "NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion". In: *IEEE Transactions on Industrial Electronics* 65.5, pp. 4392–4400. DOI: 10.1109/TIE.2017.2764844.
Condon, Robert H, Carlos M Duarte, Kylie A Pitt, Kelly L Robinson, Cathy H Lucas, Kelly R Sutherland, Hermes W Mianzan, Molly Bogeberg, Jennifer E Purcell, Mary Beth Decker, Shin-ichi Uye, Laurence P Madin, Richard D Brodeur, Steven H D Haddock, Alenka Malej, Gregory D Parry, Elena Eriksen, Javier Quinones, Marcelo Acha, Michel Harvey, James M Arthur, and William M Graham (2013). "Recurrent jellyfish blooms are a consequence of global oscillations". In: *Proceedings of the National Academy of Sciences of the United States of America* 110.3, pp. 1000–1005.
Coro, Gianpaolo and Matthew Bjerregaard Walsh (2021). "An intelligent and cost-effective remote underwater video device for fish size monitoring". In: *Ecological Informatics* 63, p. 101311. ISSN: 1574-9541. DOI: https://doi.org/10.1016/j.ecoinf.2021.101311. URL: https://www.sciencedirect.com/science/article/pii/S1574954121001023.
Dai, Jialun, Ruchen Wang, Haiyong Zheng, Guangrong Ji, and Xiaoyan Qiao (2016). "ZooplanktoNet: Deep convolutional network for zooplankton classification". In: *OCEANS 2016 - Shanghai*, pp. 1–6. DOI: 10.1109/OCEANSAP.2016.7485680.
Del Vecchio, Silvia, Edy Fantinato, Giulia Silan, and Gabriella Buffa (2018). "Trade-offs between sampling effort and data quality in habitat monitoring". In: *Biodiversity and Conservation* 28.1, pp. 55–73.
Denos, Killian, Mathieu Ravaut, Antoine Fagette, and Hock-Siong Lim (2017). "Deep learning applied to underwater mine warfare". In: *OCEANS 2017 - Aberdeen*, pp. 1–7. DOI: 10.1109/OCEANSE.2017.8084910.
Diaz-Almela, E. and C. Duarte (2008). *Management of Natura 2000 Habitats 1120, (Posidonia Oceanicae)*. Tech. rep. European Commission.
EEA (Nov. 2020). *Europe's seas and coasts*. https://www.eea.europa.eu/themes/water/europes-seas-and-coasts. Accessed: Sept. 2022.
Fenner, Peter J., John Lippmann, and Lisa-Ann Gershwin (Mar. 2010). "Fatal and Nonfatal Severe Jellyfish Stings in Thai Waters". In: *Journal of Travel Medicine* 17.2, pp. 133–138. ISSN: 1195-1982. URL: https://academic.oup.com/jtm/article-lookup/doi/10.1111/j.1708-8305.2009.00390.x.
Gao, Le, Xiaofeng Li, Fanzhou Kong, Rencheng Yu, Yuan Guo, and Yibin Ren (2022). "AlgaeNet: A Deep-Learning Framework to Detect Floating Green Algae From Optical and SAR Imagery". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15, pp. 2782–2796. DOI: 10.1109/JSTARS.2022.3162387.
Girshick, Ross (2015). "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.
Gonzalez-Cid, Yolanda, Francisco Bonin-Font, Eric Guerrrero Font, Antoni Martorell Torres, Miguel Martin Abadal, Gabriel Oliver Codina, Hilmar Hinz, Laura Pereda Briones, and Fiona Tomas (2021). "Autonomous Marine Vehicles and CNN: Tech Tools for Posidonia Meadows Monitoring". In: *OCEANS 2021: San Diego – Porto*, pp. 1–8. DOI: 10.23919/OCEANS44145.2021.9705792.
Gonzalez-Cid, Yolanda, Antoni Burguera, Francisco Bonin-Font, and Alejandro Matamoros (2017). "Machine learning and deep learning strategies to identify Posidonia meadows in underwater images". In: *OCEANS 2017 - Aberdeen*, pp. 1–5. DOI: 10.1109/OCEANSE.2017.8084991.
González-Ortegón, Enrique and Javier Moreno-Andrés (2021). "Anthropogenic Modifications to Estuaries Facilitate the Invasion of Non-Native Species". In: *Processes* 9.5. ISSN: 2227-9717. DOI: 10.3390/pr9050740. URL: https://www.mdpi.com/2227-9717/9/5/740.
Guerrero, Eric, Francisco Bonin-Font, and Gabriel Oliver (2021). "Adaptive Visual Information Gathering for Autonomous Exploration of Underwater Environments". In: *IEEE Access* 9, pp. 136487–136506. DOI: 10.1109/ACCESS.2021.3117343.

Guerrero-Font, Eric, Francisco Bonin-Font, **Miguel Martin-Abadal**, Yolanda Gonzalez-Cid, and Gabriel Oliver-Codina (2021b). "Sparse Gaussian process for online seagrass semantic mapping". In: *Expert Systems with Applications* 170, p. 114478. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.114478. URL: https://www.sciencedirect.com/science/article/pii/S095741742031126X.

Hassan, Rashid, Robert Scholes, Neville Ash, Millennium Condition, and Trends Group (Jan. 2005). *Ecosystems and Human Well-Being: Current State and Trends: Findings of the Condition and Trends Working Group (Millennium Ecosystem Assessment Series)*. Island Press.

Hays, Graeme C., Thomas K. Doyle, and Jonathan D.R. Houghton (2018). "A Paradigm Shift in the Trophic Importance of Jellyfish?" In: *Trends in Ecology and Evolution* 33 (11), pp. 874–884. ISSN: 01695347. DOI: 10.1016/j.tree.2018.09.001.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

Heshmati-Alamdari, Shahab, Charalampos P. Bechlioulis, George C. Karras, Alexandros Nikou, Dimos V. Dimarogonas, and Kostas J. Kyriakopoulos (2018). "A robust interaction control approach for underwater vehicle manipulator systems". In: *Annual Reviews in Control* 46, pp. 315 –325. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2018.10.003.

Huang, Xudong, Biao Zhang, William Perrie, Yingcheng Lu, and Chen Wang (2022). "A novel deep learning method for marine oil spill detection from satellite synthetic aperture radar imagery". In: *Marine Pollution Bulletin* 179, p. 113666. ISSN: 0025-326X. DOI: https://doi.org/10.1016/j.marpolbul.2022.113666. URL: https://www.sciencedirect.com/science/article/pii/S0025326X22003484.

Häyhä, Tiina and Pier Paolo Franzese (2014). "Ecosystem services assessment: A review under an ecological-economic and systems perspective". In: *Ecological Modelling* 289, pp. 124–132. ISSN: 0304-3800. DOI: https://doi.org/10.1016/j.ecolmodel.2014.07.002. URL: https://www.sciencedirect.com/science/article/pii/S0304380014003299.

ISA (2020). *Protection of the Marine Environment*. https://isa.org.jm/our-work/protection-marine-environment. Accessed: Sept. 2022.

Jacobi, M. and D. Karimanzira (2013). "Underwater pipeline and cable inspection using autonomous underwater vehicles". In: *2013 MTS/IEEE OCEANS - Bergen*, pp. 1–6. DOI: 10.1109/OCEANS-Bergen.2013.6608089.

Juliani, Cyril and Eric Juliani (2021). "Deep learning of terrain morphology and pattern discovery via network-based representational similarity analysis for deep-sea mineral exploration". In: *Ore Geology Reviews* 129, p. 103936. ISSN: 0169-1368. DOI: https://doi.org/10.1016/j.oregeorev.2020.103936. URL: https://www.sciencedirect.com/science/article/pii/S0169136820311215.

Kartal, Mesut and Osman Duman (2019). "Ship Detection from Optical Satellite Images with Deep Learning". In: *2019 9th International Conference on Recent Advances in Space Technologies (RAST)*, pp. 479–484. DOI: 10.1109/RAST.2019.8767844.

Kremen, Claire (2005). "Managing ecosystem services: what do we need to know about their ecology?" In: *Ecology Letters* 8.5, pp. 468–479. DOI: https://doi.org/10.1111/j.1461-0248.2005.00751.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1461-0248.2005.00751.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1461-0248.2005.00751.x.

Küpper, Frithjof C. and Nicholas A. Kamenos (2018). "The future of marine biodiversity and marine ecosystem functioning in UK coastal and territorial waters (including UK Overseas Territories) – with an emphasis on marine macrophyte communities". In: *Botanica Marina* 61.6, pp. 521–535. DOI: doi:10.1515/bot-2018-0076. URL: https://doi.org/10.1515/bot-2018-0076.

Lamb, Philip D., Ewan Hunter, John K. Pinnegar, Thomas K. Doyle, Simon Creer, Martin I. Taylor, and Marta Coll (Dec. 2019). "Inclusion of jellyfish in 30+ years of Ecopath with Ecosim models". In: *ICES Journal of Marine Science* 76 (7), pp. 1941–1950. ISSN: 10959289. DOI: 10.1093/icesjms/fsz165.

Lee, JH, Choi HW, J Chae, DS Kim, and SB Lee (2006). "Performance analysis of intake screens in power plants on mass impingement of marine organisms". In: *Ocean and polar research* 28, pp. 385–393.

Li, Daoliang and Ling Du (June 2022). "Recent advances of deep learning algorithms for aquacultural machine vision systems with emphasis on fish". In: *Artificial Intelligence Review* 55. DOI: 10.1007/s10462-021-10102-3.

Li, Xiu, Min Shang, Hongwei Qin, and Liansheng Chen (2015). "Fast accurate fish detection and recognition of underwater images with Fast R-CNN". In: *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–5. DOI: 10.23919/OCEANS.2015.7404464.

Li, Yan, Jiahong Guo, Xiaomin Guo, Zhiqiang Hu, and Yu Tian (2021). "Plankton Detection with Adversarial Learning and a Densely Connected Deep Learning Model for Class Imbalanced Distribution". In: *Journal of Marine Science and Engineering* 9.6. ISSN: 2077-1312. DOI: 10.3390/jmse9060636. URL: https://www.mdpi.com/2077-1312/9/6/636.

Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg (2016). "SSD: Single Shot MultiBox Detector". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, pp. 21–37. ISBN: 978-3-319-46448-0.

Lohia, Aditya, Kalyani Kadam, Rahul Joshi, and Dr Bongale (2021). "Bibliometric Analysis of One-stage and Two-stage Object Detection". In: *Library Philosophy and Practice*. URL: https://digitalcommons.unl.edu/libphilprac/4910/. (Accessed: Sept. 2022).

Marba, Nuria and Carlos Duarte (2010). "Mediterranean warming triggers seagrass (Posidonia oceanica) shoot mortality". English. In: *Global Change Biology* 16.8, pp. 2366–2375. ISSN: 1354-1013.

Maurer, Brian A. (2009). "Ecological complexity". In: *Encyclopedia of Complexity and Systems Science*. Ed. by Robert A. Meyers. New York, NY: Springer New York, pp. 2697–2711. ISBN: 978-0-387-30440-3. DOI: 10.1007/978-0-387-30440-3_162. URL: https://doi.org/10.1007/978-0-387-30440-3_162.

Mohamed, Hassan, Kazuo Nadaoka, and Takashi Nakamura (2022). "Automatic Semantic Segmentation of Benthic Habitats Using Images from Towed Underwater Camera in a Complex Shallow Water Environment". In: *Remote Sensing* 14.8. ISSN: 2072-4292. DOI: 10.3390/rs14081818. URL: https://www.mdpi.com/2072-4292/14/8/1818.

Morales, Eduardo, Rafael Murrieta-Cid, Israel Becerra, and Marco Esquivel Basaldua (Nov. 2021). "A survey on deep learning and deep reinforcement learning in robotics with a tutorial on deep reinforcement learning". In: *Intelligent Service Robotics* 14. DOI: 10.1007/s11370-021-00398-z.

Nassif, Ali Bou, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan (2019). "Speech Recognition Using Deep Neural Networks: A Systematic Review". In: *IEEE Access* 7, pp. 19143–19165. DOI: 10.1109/ACCESS.2019.2896880.

Nayak, Nandeeka, Makoto Nara, Timmy Gambin, Zoë Wood, and Christopher M. Clark (2021). "Machine Learning Techniques for AUV Side-Scan Sonar Data Feature Extraction as Applied to Intelligent Search for Underwater Archaeological Sites". In: *Field and Service Robotics*. Ed. by Genya Ishigami and Kazuya Yoshida. Singapore: Springer Singapore, pp. 219–233. ISBN: 978-981-15-9460-1.

Nguyen, Huu-Thu, Eon-Ho Lee, and Sejin Lee (2020). "Study on the Classification Performance of Underwater Sonar Image Classification Based on Convolutional Neural Networks for Detecting a Submerged Human Body". In: *Sensors* 20.1. ISSN: 1424-8220. DOI: 10.3390/s20010094. URL: https://www.mdpi.com/1424-8220/20/1/94.

Norgaard, Richard B. (2010). "Ecosystem services: From eye-opening metaphor to complexity blinder". In: *Ecological Economics* 69.6. Special Section - Payments for Environmental Services: Reconciling Theory and Practice, pp. 1219–1227. ISSN: 0921-8009. DOI: https://doi.org/10.1016/j.ecolecon.2009.11.009. URL: https://www.sciencedirect.com/science/article/pii/S0921800909004583.

Otter, Daniel W., Julian R. Medina, and Jugal K. Kalita (2021). "A Survey of the Usages of Deep Learning for Natural Language Processing". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.2, pp. 604–624. DOI: 10.1109/TNNLS.2020.2979670.

Park, Jungsu, Jiwon Baek, Jongrack Kim, Kwangtae You, and Keugtae Kim (2022). "Deep Learning-Based Algal Detection Model Development Considering Field Application". In: *Water* 14.8. ISSN: 2073-4441. DOI: 10.3390/w14081275. URL: https://www.mdpi.com/2073-4441/14/8/1275.

Pitt, Kylie A., Cathy H. Lucas, Robert H. Condon, Carlos M. Duarte, and Ben Stewart-Koster (Nov. 2018). "Claims That Anthropogenic Stressors Facilitate Jellyfish Blooms Have Been Amplified Beyond the Available Evidence: A Systematic Review". In: *Frontiers in Marine Science* 5. ISSN: 22967745. DOI: 10.3389/fmars.2018.00451.

Pizarro, Oscar, Ariell Friedman, Mitch Bryson, Stefan B. Williams, and Joshua Madin (2017). "A simple, fast, and repeatable survey method for underwater visual 3D benthic mapping and monitoring". In: *Ecology and Evolution* 7.6, pp. 1770–1782. DOI: https://doi.org/10.1002/ece3.2701. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ece3.2701. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.2701.

Politikos, D., Elias Fakiris, Athanasios Davvetas, Iraklis Klampanos, and George Papatheodorou (Mar. 2021). "Automatic detection of seafloor marine litter using towed camera images and deep learning". In: *Marine Pollution Bulletin* 164, p. 111974. DOI: 10.1016/j.marpolbul.2021.111974.

Purcell, J. E., E. J. Baxter, and V. L. Fuentes (2013). "Jellyfish as products and problems of aquaculture". In: *Advances in Aquaculture Hatchery Technology*, pp. 404–430. ISSN: 0966-0461. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780857091192500139.

Py, Ouyang, Hu Hong, and Shi Zhongzhi (2016). "Plankton classification with deep convolutional neural networks". In: *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pp. 132–136. DOI: 10.1109/ITNEC.2016.7560334.

Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng (Jan. 2009). "ROS: an open-source Robot Operating System". In: *ICRA Workshop on Open Source Software*. Vol. 3.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi (2016). "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. DOI: 10.48550/ARXIV.1506.01497. URL: https://arxiv.org/abs/1506.01497.

Richardson, Anthony J, Andrew Bakun, Graeme C Hays, and Mark J Gibbons (2009). "The jellyfish joyride: causes, consequences and management responses to a more gelatinous future". In: *Trends in Ecology & Evolution* 24.6, pp. 312–322.

Ridao, Pere, Marc Carreras, David Ribas, Pedro J. Sanz, and Gabriel Oliver (Nov. 2015). "Intervention AUVs: The Next Challenge". In: *Annual Reviews in Control* 40, pp. 227–241. DOI: 10.1016/j.arcontrol.2015.09.015.

Ruiz-Frau, Ana, **Martin-Abadal, Miguel**, Charlotte L. Jennings, Yolanda Gonzalez-Cid, and Hilmar Hinz (2022). "The potential of Jellytoring 2.0 smart tool as a global jellyfish monitoring platform". In: *Ecology and Evolution* 12.11. e9472 ECE-2022-04-00522.R2, e9472. DOI: https://doi.org/10.1002/ece3.9472.

Simonyan, Karen and Andrew Zisserman (Sept. 2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv 1409.1556*.

SRV (2018). *Project webpage for "Real-time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks"*. http://srv.uib.es/twinbot-twin-robots-for-cooperative-underwater-intervention-missions/. Accessed: Sept. 2022.

— (2020). *Project webpage for "Real-time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks"*. http://srv.uib.es/detecpos/. Accessed: Sept. 2022.

— (2021). *Project webpage for "Real-time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks"*. http://srv.uib.es/project-cooperamos-subproject-vi-smart/. Accessed: Sept. 2022.

St. John, Michael A., Angel Borja, Guillem Chust, Michael Heath, Ivo Grigorov, Patrizio Mariani, Adrian P. Martin, and Ricardo S. Santos (2016). "A Dark Hole in Our Understanding of Marine Ecosystems and Their Services: Perspectives from the Mesopelagic Community". In: *Frontiers in Marine Science* 3. ISSN: 2296-7745. DOI: 10.3389/fmars.2016.00031. URL: https://www.frontiersin.org/articles/10.3389/fmars.2016.00031.

Strong, James Asa, Eider Andonegi, Kemal Can Bizsel, Roberto Danovaro, Mike Elliott, Anita Franco, Esther Garces, Sally Little, Krysia Mazik, Snejana Moncheva, Nadia Papadopoulou, Joana Patrício, Ana M. Queirós, Chris Smith, Kremena Stefanova, and Oihana Solaun (2015). "Marine biodiversity and ecosystem function relationships: The potential for practical monitoring applications". In: *Estuarine, Coastal and Shelf Science* 161, pp. 46–64. ISSN: 0272-7714. DOI: https://doi.org/10.1016/j.ecss.2015.04.008. URL: https://www.sciencedirect.com/science/article/pii/S0272771415001389.

Szegedy, Christian, Sergey Ioffe, and Vincent Vanhoucke (Feb. 2016). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *AAAI Conference on Artificial Intelligence*.

Tang, Gang, Shibo Liu, Iwao Fujino, Christophe Claramunt, Yide Wang, and Shaoyang Men (2020). "H-YOLO: A Single-Shot Ship Detection Approach Based on Region of Interest Preselected Network". In: *Remote Sensing* 12.24. ISSN: 2072-4292. DOI: 10.3390/rs12244192. URL: https://www.mdpi.com/2072-4292/12/24/4192.

Telesca, Luca, Andrea Belluscio, Alessandro Criscoli, Giandomenico Ardizzone, Eugenia T. Apostolaki, Simonetta Fraschetti, Michele Gristina, Leyla Knittweis, Corinne S. Martin, Gérard Pergent, Adriana Alagna, Fabio Badalamenti, Germana Garofalo, Vasilis Gerakaris, Marie Louise Pace, Christine Pergent-Martini, and Maria Salomidi (2015). "Seagrass meadows (Posidonia oceanica) distribution and trajectories of change". In: *Scientific reports*.

Martin-Abadal, Miguel (2018). *Posidonia oceanica Segmentation*. https://github.com/srv/Posidonia-semantic-segmentation.

Martin-Abadal, Miguel (2020). *Jellyfish Object Detection*. https://github.com/srv/jf_object_detection.

Martin-Abadal, Miguel, Gabriel Oliver-Codina, and Yolanda Gonzalez-Cid (2022a). *Project webpage for "Real-time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks"*. http://srv.uib.es/3d-pipes-2/. Accessed: Sept. 2022.

— (2022b). "Real-Time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks". In: *Sensors* 22.21. ISSN: 1424-8220. DOI: 10.3390/s22218141.

Martin-Abadal, Miguel, Manuel Piñar-Molina, Antoni Martorell-Torres, Gabriel Oliver-Codina, and Yolanda Gonzalez-Cid (2021a). *Project webpage for "Underwater Pipe and Valve 3D Recognition Using Deep Learning Segmentation"*. http://srv.uib.es/3d-pipes-1/. Accessed: Sept. 2022.

Thum, Guan Wei, Sai Hong Tang, Siti Azfanizam Ahmad, and Moath Alrifaey (2020). "Toward a Highly Accurate Classification of Underwater Cable Images via Deep Convolutional Neural Network". In: *Journal of Marine Science and Engineering* 8.11. ISSN: 2077-1312. DOI: 10.3390/jmse8110924. URL: https://www.mdpi.com/2077-1312/8/11/924.

Villon, Sébastien, Marc Chaumont, Gérard Subsol, Sébastien Villéger, Thomas Claverie, and David Mouillot (2016). "Coral Reef Fish Detection and Recognition in Underwater Videos by Supervised Machine Learning: Comparison Between Deep Learning and HOG+SVM Methods". In: *Advanced Concepts for Intelligent Vision Systems*. Ed. by Jacques Blanc-Talon, Cosimo Distante, Wilfried Philips, Dan Popescu, and Paul Scheunders. Cham: Springer International Publishing, pp. 160–171. ISBN: 978-3-319-48680-2.

Wang, Yue, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon (Oct. 2019). "Dynamic Graph CNN for Learning on Point Clouds". In: *ACM Trans. Graph.* 38.5. ISSN: 0730-0301. DOI: 10.1145/3326362.

Yang, Yi-Jie, Suman Singha, and Roberto Mayerle (2022). "A deep learning based oil spill detector using Sentinel-1 SAR imagery". In: *International Journal of Remote Sensing* 43.11, pp. 4287–4314. DOI: 10.1080/01431161.2022.2109445. eprint: https://doi.org/10.1080/01431161.2022.2109445. URL: https://doi.org/10.1080/01431161.2022.2109445.

Yu, Mengxi, Joshiba Ariamuthu Venkidasalapathy, Yueqi Shen, Noor Quddus, and M. Sam Mannan (Jan. 2017). "Bow-tie Analysis of Underwater Robots in Offshore Oil and Gas Operations". In: *Offshore Technology Conference*. DOI: 10.4043/27818-MS.

Zhang, Hui, Yonglin Tian, Kunfeng Wang, Wensheng Zhang, and Fei-Yue Wang (2020). "Mask SSD: An Effective Single-Stage Approach to Object Instance Segmentation". In: *IEEE Transactions on Image Processing* 29, pp. 2078–2093. DOI: 10.1109/TIP.2019.2947806.