Politecnico di Torino

Master's Thesis in

Environmental and Territorial Engineering

Deep Learning-Based Multitemporal Mapping of *Posidonia oceanica* from High-Resolution Multispectral Satellite Imagery

Candidate: Payam Yari

Supervisors

Prof. Andrea Maria Lingua
Arch. Francesca Matrone

Co-supervisors

Ing. Stefania Manca
Ing. Francesca Gallitto

October 9, 2025

Abstract

Seagrass meadows of *Posidonia oceanica (P.O.)* are key coastal habitats in the Mediterranean, yet they remain difficult and costly to survey frequently with traditional field methods. This thesis develops and evaluates a remote sensing and deep learning workflow that enables routine, spatially explicit mapping of *P.O.* from freely available satellite imagery. The work aligns with the broader aims of the POSEIDON initiative to support European directives through reproducible, non-destructive, georeferenced monitoring of priority marine habitats.

The study focuses on the Capo Testa Marine Protected Area in northern Sardinia. Level-2A Sentinel-2 multispectral images from 2015 to 2024 were processed and used to train semantic segmentation models. Two architectures were considered, U-Net and DeepLabv3, each coupled with ResNet encoders. Model performance was assessed on held-out data. The resulting classifier achieved an overall accuracy of about 88% for discriminating *P.O.* from spectrally similar substrates and water.

Multi-temporal classification and change detection analysis indicate a net loss of about 8% in meadow extent between 2015 and 2024, with spatially heterogeneous patterns that are consistent with a combination of local anthropogenic pressures (for example anchoring and coastal works) and climate-related stressors (warming, turbidity variability, and storm exposure).

The proposed pipeline is cost-effective in terms of time, scalable, and transferable to other Mediterranean sites. It can underpin operational monitoring and planning while reducing dependence on intensive field campaigns. The study also highlights the value of targeted ground truth through field visits, underwater photography, and photogrammetry to calibrate and validate models and to refine change attribution.

Keywords: Posidonia Oceanica; Remote Sensing; Sentinel-2; semantic segmentation; Deep Learning; U-Net; DeepLabv3; ResNet; change detection.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 *Posidonia oceanica*

P.O. is a marine flowering plant belonging to the family Posidoniaceae, and is strictly endemic to the Mediterranean Sea. Unlike algae, it is a true phanerogam with roots, rhizomes, stems, leaves, flowers, and fruits (Hemminga et al., 2000; Damiana, 2023). This species forms extensive underwater meadows, sometimes referred to as the "forests of the Mediterranean," that can persist for thousands of years due to their very slow growth rate (on the order of centimeters per year) (Telesca et al., 2015).

The morphology of Posidonia is characterized by ribbon shaped leaves, which can reach up to 1 to 2 meters in length, and a complex rhizome system. Horizontal rhizomes spread laterally, anchoring the plant through roots, while vertical rhizomes accumulate organic matter and form the so-called *matte*, a dense biogenic structure made of intertwined living and dead rhizomes that stabilizes sediments and modifies the sea bottom (Pergent et al., 1994; Boudouresque et al., 2009) (Figure 1.1). Reproduction occurs primarily through clonal growth, but flowering and fruiting events have been increasingly recorded in recent decades, possibly stimulated by warming trends (Danovaro et al., 2019).

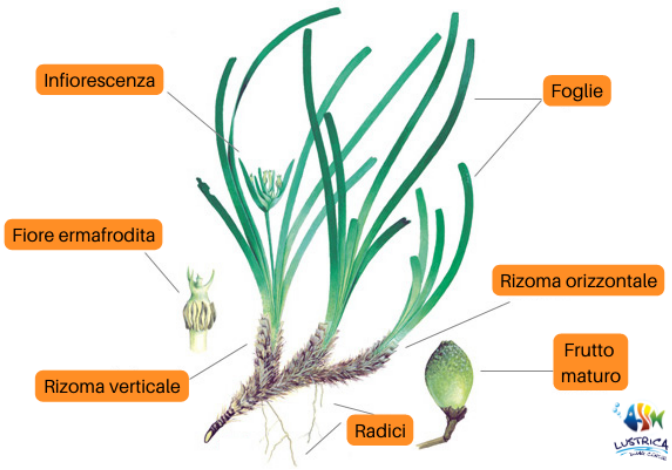


Figure 1.1: Morphological structure of P.O., showing roots, rhizomes, leaves, flowers, and fruit (Damiana 2023).

Posidonia meadows colonize both rocky and sandy substrates, from the surface down to depths exceeding 40-50 meters, depending on the clarity of the water. Due to their ecological significance, they are recognized as habitats of community importance under the EU Habitats Directive (92/43/EEC) and benefit from strict protection under multiple Mediterranean conventions (Danovaro et al., 2019).

From an ecological perspective, these meadows are considered the Mediterranean equivalent of coral reefs in tropical seas: they provide habitat and nursery grounds for numerous species, produce large amounts of oxygen (up to 15 L O_2 per m^2 per day), and act as a major sink of carbon dioxide (Boudouresque et al., 2017; Waycott et al., 2009). The dense leaf canopy buffers the wave energy, while the matte structure prevents coastal erosion and contributes to the stability of the coastline (Figure 1.2). Even dead leaves that accumulate as banquettes along beaches continue to play an ecological role, protecting beaches from erosion and offering habitat for small organisms.

Figure 1.2: P.O. meadow and ecological role in the Mediterranean (adapted from Danovaro et al. 2019).

1.2 Background and motivation

P.O. forms dense underwater meadows that rank among the most representative and ecologically significant ecosystems of the Mediterranean due to their spatial extent, structural complexity, and exceptional longevity (Buia et al., 2004). Beyond their ecological role, these meadows are crucial for coastal engineering and management because of the services they provide to human societies.

From an engineering perspective, P.O. helps stabilize sediments, attenuate wave energy, and regulate water quality. The meadows act as natural filters for suspended particles, produce oxygen, and serve as efficient carbon sinks. In fact, they sequester carbon dioxide at rates up to ten times higher than temperate forests and fifty times higher than tropical forests on an equivalent area basis (Buia et al., 2004). These features make Posidonia a key ally in efforts to mitigate climate change and preserve coastal resilience.

1.3 Ecological and environmental significance

The ecological functions of Posidonia meadows extend well beyond physical protection of coastlines. They support a remarkably rich biodiversity, serving as breeding and nursery grounds for more than 1,400 marine species, ranging from invertebrates to commercially valuable fish (Boudouresque et al., 2006). The persistence of such ecosystems supports fisheries, tourism, and overall coastal productivity.

Due to its long lifespan and extremely slow growth rate, *P.O.* is highly sensitive to environmental disturbances. Even small-scale damage can require decades to recover. These characteristics make it a valuable biological indicator of ecosystem health, and its importance has been formally acknowledged in European and Mediterranean conservation frameworks, including the Barcelona Convention (Lopez y Royo et al., 2010).

1.4 Impacts of climate change and anthropogenic stressors

Despite their ecological importance and natural resilience, Posidonia meadows are increasingly vulnerable to human and climate pressures. Anthropogenic stressors include mechanical damage caused by anchoring, dredging, and trawling; chronic pollution from coastal settlements and industry; increased turbidity due to sediment resuspension; and habitat fragmentation linked to urbanization (Boudouresque et al., 2006).

On top of these pressures, climate change intensifies existing vulnerabilities. Rising sea surface temperatures can affect photosynthetic efficiency and trigger mortality during prolonged marine heat waves. Ocean acidification has potential effects on seagrass epiphyte interactions. Altered storm regimes increase physical disturbance, while sea level rise and coastal erosion may shift the depth range of meadows, forcing regression at their deeper distribution limits. In some regions of the Mediterranean, annual meadow loss has been estimated at up to 5% (Francour et al., 1999; Milazzo et

al., 2004), a rate that far exceeds the natural growth capacity of this species. Without effective protection and restoration, such trends could lead to irreversible ecological damage.

1.5 The need for monitoring and assessment

Given these threats, accurate and regular monitoring of Posidonia meadows is essential. Traditional field surveys, although highly accurate at local scales, are labor intensive, costly, and limited in coverage. In contrast, remote sensing offers a scalable solution, enabling the observation of seagrass dynamics across regional and decadal time scales. Satellite platforms such as Sentinel-2 and Landsat provide multispectral imagery with sufficient spatial and temporal resolution to support systematic monitoring.

Recent advances in computational techniques further enhance these opportunities. Deep learning models, particularly convolutional neural networks (CNNs) and architectures such as U-Net and DeepLab, have demonstrated remarkable success in semantic segmentation tasks. Their application to benthic habitat mapping enables precise classification of submerged vegetation, quantification of temporal changes, and ultimately, data driven management of marine protected areas. For environmental engineers and planners, these tools provide a pathway toward more proactive and sustainable coastal management strategies.

1.6 Aim and structure of the thesis

The overarching aim of this thesis is to develop and apply a deep learning-based framework for the classification and multitemporal assessment of P.O. meadows using high-resolution multispectral satellite imagery. The study focuses on selected coastal sites in southern Italy, areas where the species remains widespread but is exposed to significant human and climatic pressures.

The specific objectives are:

1. Preprocess and harmonize Sentinel-2 imagery for selected coastal sites.
2. Train and validate deep learning models for accurate classification of Posidonia meadows.
3. Quantify decadal-scale changes in meadow extent between 2015 and 2024.
4. Interpret these changes within the broader context of climate stressors and anthropogenic impacts.

The remainder of this thesis is structured as follows:

- Chapter 2 reviews the ecological role of Posidonia, remote sensing approaches, and applications of deep learning in environmental monitoring.
- Chapter 3 introduces the study areas, datasets, and preprocessing workflow.
- Chapter 4 details the methodology for deep learning classification and validation.
- Chapter 5 presents the results of classification and change detection, interpreting them in light of ecological and climate pressures.
- Chapter 6 concludes with the main findings, implications for conservation and management, and recommendations for future research.

Chapter 2: Literature Review

2.1 Introduction to the literature review

The purpose of this chapter is to situate the present research within the broader scientific context of monitoring and mapping *P.O.* meadows. While the ecological importance and vulnerability of these ecosystems have been outlined in Chapter 1, it is now necessary to examine how previous studies have approached their observation, mapping, and long-term assessment. In particular, the literature highlights both the urgency of protecting these habitats and the technological innovations that enable large-scale monitoring.

This review is structured around four thematic areas. The first section synthesizes previous studies on *P.O.*, with particular emphasis on patterns of distribution, reported decline, and the motivations for systematic spatial monitoring. The second section examines the use of remote sensing technologies for seagrass mapping, tracing the progression from early aerial and sonar surveys to the current use of multispectral and hyperspectral satellite platforms. The third section explores the development of deep learning techniques for semantic segmentation, highlighting their advantages over traditional classification methods in environmental applications. Finally, the fourth section reviews research that integrates satellite imagery with deep learning approaches for seagrass habitat classification, with a specific focus on Mediterranean case studies.

Together, these perspectives provide a foundation for understanding the state of the art in seagrass mapping and justify the methodological choices adopted in this thesis. By synthesizing ecological insights with engineering-relevant monitoring approaches, the review highlights how modern computational methods can support more accurate, efficient, and scalable assessments of *P.O.* meadows.

2.2 Studies on *P.O.*

Research on *P.O.* has extended well beyond biology, increasingly addressing the need for systematic observation and spatial monitoring. European legislation, most notably the Habitats Directive (Council Directive 92/43/EEC) and the Water Framework Directive,

provides a regulatory foundation that has spurred regional and national initiatives to map seagrass meadows and assess their conservation status (Montefalcone, 2009).

Scientific studies have documented that disturbances such as anchoring, dredging, and the spread of invasive species (e.g., *Caulerpa taxifolia*) are linked to meadow regression across various Mediterranean sites (Boudouresque et al., 2006). More recent analyses also point to the influence of sediment extraction and rising sea surface temperatures, which can exacerbate local losses. While these findings originate primarily from ecological research, they underline the necessity of reliable, spatially explicit information on meadow distribution.

For coastal engineers and environmental planners, such data are indispensable in marine spatial planning, environmental impact assessments, and habitat restoration efforts. Because traditional field-based surveys are labor intensive and spatially limited, growing attention has been directed toward satellite-based mapping as a complementary and scalable approach. This methodological shift is central to the present thesis, which builds on these earlier insights by applying advanced computational tools to monitor changes in *P.O.* meadows.

2.3 Remote sensing for seagrass mapping

2.3.1 General applications of remote sensing

Remote sensing (RS) has substantially advanced the monitoring of submerged vegetation by enabling spatially extensive and repeatable observations. Early seagrass mapping relied on Landsat and IKONOS imagery, later complemented by higher-resolution platforms such as WorldView-2/3 and the free, high-revisit Sentinel-2 constellation. Methodological baselines include depth-invariant transforms and water-column corrections to stabilize benthic reflectance (Sagawa et al., 2010), while recent systematic reviews emphasize that Sentinel-2 dominates contemporary seagrass monitoring and Landsat remains essential for multi-decadal change analysis (Campillo-Tamarit et al., 2025). In practice, supervised machine learning approaches (e.g., Random Forest, SVM) and spectral indices (e.g., NDVI, NDWI, EVI) are widely used, with performance contingent on water clarity, depth, and the integration of ancillary datasets such as bathymetry and turbidity (Huete, 2004; Mederos-Barrera et al., 2022).

2.3.2 Applications to *P.O.*

Because *P.O.* is endemic to the Mediterranean, many RS-based studies have focused on this region. Using Landsat 8 OLI, (Borfecchia et al., 2019) mapped meadows

along Mediterranean coasts, while (D. Traganos et al., 2018) demonstrated robust detection with Sentinel-2 MSI, highlighting the benefits of high revisit frequency and visible-to-near-infrared spectral bands. Higher spatial resolution further improves boundary delineation: (Dattola et al., 2018) compared Sentinel-2 and Landsat-8 with MIVIS and WorldView-2, showing finer delineation of meadows, and (Topouzelis et al., 2018) integrated OBIA with Landsat-8 to refine seagrass mapping in Greek waters. Very high-resolution imagery has also enabled the delineation of deep limits, with (Poursanidis et al., 2018) mapping meadows to 38-45 m depth in Crete at accuracies above 90%.

For long-term perspectives, (Telesca et al., 2015) analyzed meadow status and pressures across Mediterranean basins, and (Cingano et al., 2024) applied two decades of Landsat data with Random Forest in Italian lagoons, reporting a net increase in cover driven by *Nanozostera noltei*. Beyond Italy, (Davies et al., 2024) used a neural network classifier on Sentinel-2 time-series to track intertidal seagrass dynamics across six Western European sites, capturing distinct trajectories and phenological gradients. Other comparative studies showed that, under some conditions, machine learning applied to uncorrected imagery can outperform traditional water-column correction (Mederos-Barrera et al., 2022). Together, these studies demonstrate the potential of multispectral archives to capture both spatial extent and temporal dynamics of *P.O.* and other seagrasses.

2.4 Deep learning for semantic segmentation and environmental mapping

Deep learning (DL) has become a cornerstone of modern image analysis, offering significant advantages over traditional classifiers by automatically learning multiscale features from raw data. In the context of environmental monitoring, its most relevant application is semantic segmentation, the classification of each pixel into a specific class. This approach enables fine-grained mapping of heterogeneous landscapes and seascapes, capturing both local details and broader contextual patterns. Architectures such as Fully Convolutional Networks, U-Net, and DeepLab have established themselves as robust tools for segmentation tasks across domains ranging from medical imaging to remote sensing (Ronneberger et al., 2015a; Chen et al., 2017a).

In marine environments, these methods have been increasingly applied to the detection and monitoring of seagrass meadows. For instance, Sentinel-2 based deep learning workflows have demonstrated rapid and accurate mapping of *P.O.* cover, enabling both short- and long-term assessments of human and climatic pressures (Chowdhury et al., 2024). Studies using underwater imagery further show that neural

networks can discriminate *P.O.* patches under challenging optical conditions (Burguera, 2020). These applications highlight the potential of DL to complement traditional monitoring by delivering scalable, data-driven habitat classifications.

Among the available architectures, U-Net has gained particular prominence due to its encoder–decoder structure with skip connections, which is effective for preserving spatial detail in high-resolution imagery. Variants that incorporate backbones such as ResNet or EfficientNet further enhance feature extraction capacity. Since this thesis employs a U-Net-based model for the classification of *P.O.* meadows, a more detailed description of the architecture and training strategy is provided in Chapter 4.

2.5 Integrating remote sensing and deep learning in seagrass monitoring

2.5.1 Deep learning for satellite-based seagrass Classification

Deep learning has improved both accuracy and transferability in seagrass mapping from satellite data. In Italy, (Scarpetta et al., 2022) applied U-Net to high-resolution imagery along the Apulian coast, achieving reliable pixel-wise classification of *P.O.* with limited training data. Advances in land cover segmentation also inform marine applications: (Maurya et al., 2023) developed a 33-layer modified U-Net for Sentinel-2, achieving higher IoU scores than standard U-Net and DeeplabV3+. At broader scales, (Peng et al., 2025) introduced SGDenseNet, a DenseNet-based model trained on Caribbean data and validated globally, outperforming OBIA and shallow-learning methods and demonstrating the feasibility of low-cost, generalizable seagrass products.

2.5.2 Multi-sensor integration and operational frameworks

Beyond single-sensor classification, multisensor integration provides new opportunities for seagrass monitoring. At site scale, (Rende et al., 2020) combined Pléiades imagery, UAV orthophotos, multibeam bathymetry, and underwater photogrammetry in an object-based framework, achieving accuracies above 95-99%. At operational scale, (Chowdhury et al., 2024) demonstrated a pipeline combining Sentinel-2, ACOLITE preprocessing, and DL to map meadows in the Balearic and Maltese Islands between 2017 and 2021, with accuracy up to 92% and transferability across sites. Larger benchmarking efforts in the Mediterranean have tested multiple DL architectures across multi-scene datasets, such as the CAMELE framework proposed by (Giménez-Romero et al., 2024). On a continental scale, (Dimosthenis Traganos et al., 2018) used Google Earth Engine with Sentinel-2 and SVM to map 2,510 km^2 of meadows in the Aegean and Ionian Seas, showing the feasibility of low-cost, cloud-based monitoring. Similar

continental-to-global DL frameworks continue to emerge, with examples in Western Europe (Davies et al., 2024) and the Caribbean (Peng et al., 2025). Lessons from large-area terrestrial land cover projects reinforce the importance of OBIA, ancillary layers, and scalable workflows for national and continental mapping (Maxwell et al., 2019).

2.5.2.1 Alignment with national research initiatives

A particularly relevant initiative is the POSEIDON Project (Ceccherelli et al., 2024), which is a research project of Italia Domani - Piano Nazionale di Ripresa e Resilienza (PNRR) and carried out by a consortium that includes researchers from the Politecnico di Torino. The project combines satellite archives (Landsat, Sentinel-2) with UAV imaging and acoustic surveys to deliver multi-scale, multi-temporal products for *P.O.* meadows and banquettes, including a webGIS and ecological models in support of conservation and marine spatial planning. The approach developed in this thesis leveraging Sentinel-2 and deep learning for multitemporal mapping aligns with and complements the goals of POSEIDON led by the PoliTo and other collaborators.

Chapter 3: Materials and Methods

3.1 Introduction

This chapter presents the practical methodology developed to identify and classify meadows of *P.O.* using satellite-based imagery and deep learning techniques. Building on the conceptual and technical foundations laid in Chapter 2 which outlined the ecological role of *P.O.*, the capabilities of remote sensing, and the promise of deep learning in environmental applications, this section now shifts the focus to the operational framework adopted in this study.

Given the increasing availability of high-resolution multispectral data, particularly from the Sentinel-2 satellite constellation, and the maturity of convolutional neural networks for image analysis, the workflow designed for this thesis combines both domains. This integration is particularly valuable for mapping seagrass meadows in shallow coastal areas, where underwater reflectance patterns can be highly variable due to factors such as water depth, turbidity, and seabed composition.

The methodology follows a structured sequence of phases. It begins with the selection and preprocessing of satellite images, including spatial resampling and the delineation of the areas of interest. This is followed by the design and implementation of a supervised deep learning model, based on the U-Net architecture, which has proven effective for pixel-level classification. To optimize the performance of the model, particular attention is given to training strategies that address challenges such as class imbalance, overfitting, and spectral noise. Finally, the results are evaluated using both spatial accuracy metrics and classification performance indicators.

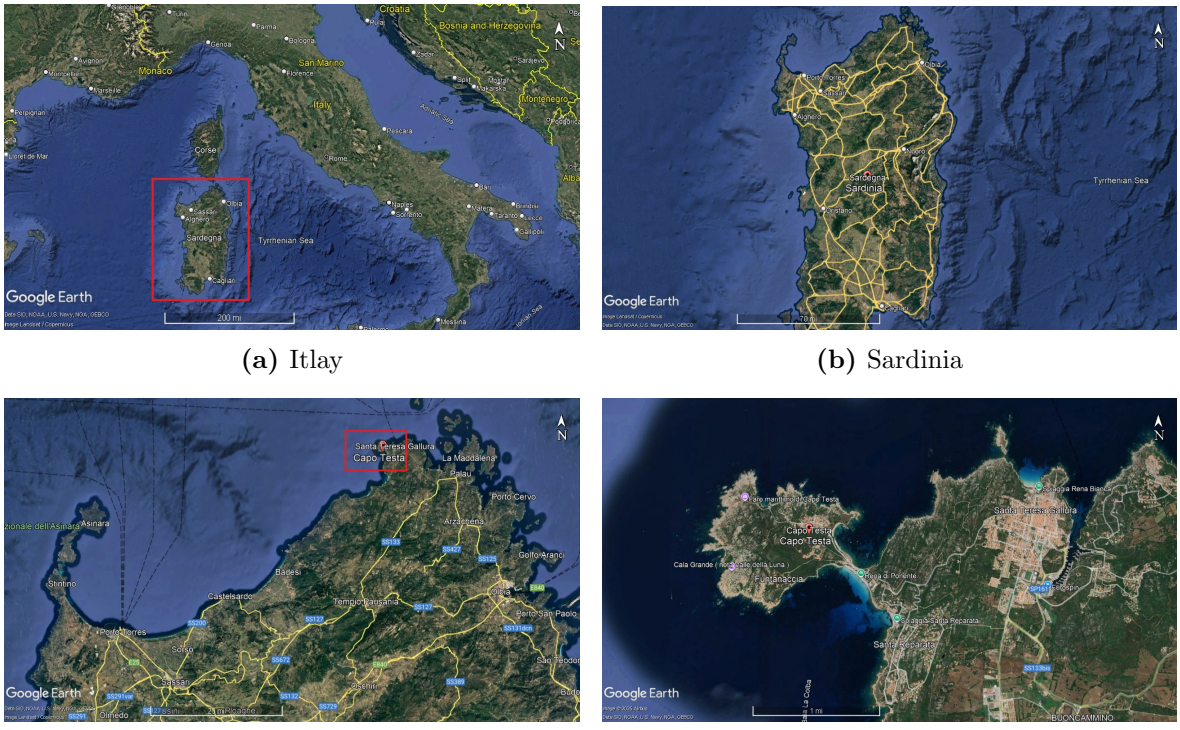
In addition, this chapter provides a detailed description of the coastal areas selected for the study: specific marine zones along the southern and northeast coasts of Sardinia, Italy, which were chosen because of their ecological importance and the availability of supporting data. These include both protected marine areas and ecologically sensitive sites, as identified by regional conservation frameworks.

Subsequent sections outline each component of the workflow, from data acquisition

to model evaluation, providing a transparent and reproducible methodology for remote sensing-based monitoring of *P.O.*. This chapter forms the technical backbone of the thesis, laying the groundwork for the analysis and discussion presented in Chapters 4 and 5.

3.2 Study areas

Sardinia, the second largest island in the Mediterranean Sea, lies just south of mainland Italy and is surrounded by the Tyrrhenian and western Mediterranean waters. Its unique geographical position, combined with a rugged coastline and extensive shallow waters, makes it one of Italy's most ecologically and economically important regions. The island is known for its rich biodiversity, pristine beaches, and a growing tourism industry that attracts millions of visitors each year. However, in addition to its natural beauty, Sardinia faces significant ecological challenges arising from climate change, recreational boating, commercial shipping, and coastal development.



(a) Itlay **(b)** Sardinia

(c) North of Sardinia **(d)** Capo Testa

Figure 3.1: Study area: Sardinia (Italy) and the Capo Testa peninsula (approx. 41.24 N, 9.14 E). Source: Google Earth; accessed Sep 2025.

In recognition of its environmental importance and the growing anthropogenic pressures, six Marine Protected Areas (MPAs) have been established in Sardinia under regional and national frameworks. These are:

- Capo Carbonara Marine Protected Area

- Tavolara – Punta Coda Cavallo Marine Protected Area
- Asinara Island Marine Protected Area
- Capo Caccia – Isola Piana Marine Protected Area
- Sinis Peninsula – Mal di Ventre Island Marine Protected Area
- Capo Testa – Punta Falcone Marine Protected Area

Figure 3.1 shows the geographical location of Sardinia in the Mediterranean Sea.

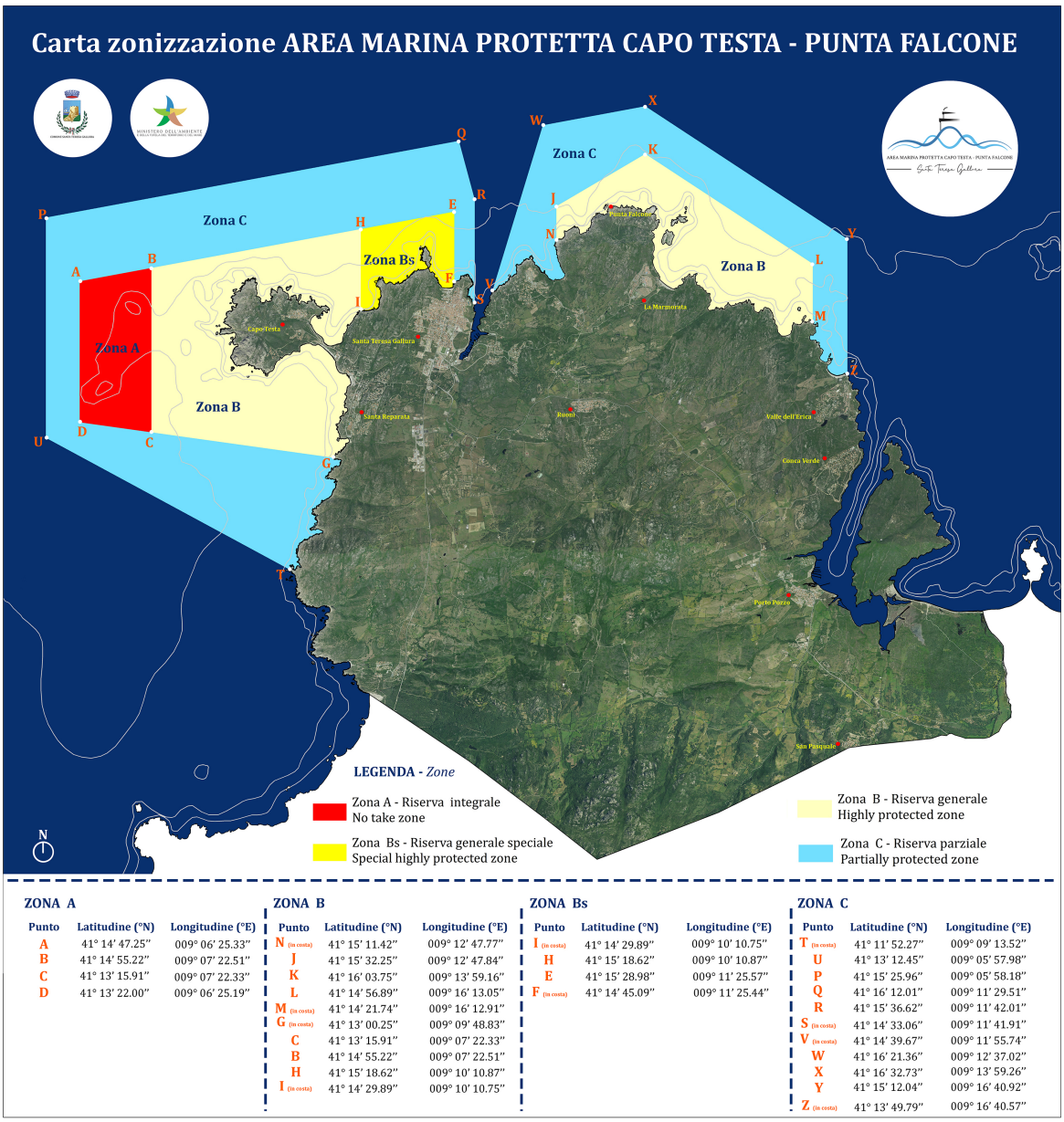
Among these, the Capo Testa – Punta Falcone Marine Protected Area, located along the northern coast of Sardinia, was selected as the primary case study for this research. Established by ministerial decree in 2018 and managed by the Municipality of Santa Teresa Gallura, this MPA spans over 5,000 hectares and encompasses a wide range of marine and coastal environments representative of the Gallura region's natural heritage (Capo Testa – Punta Falcone MPA, 2024).

This protected area is subdivided into four zones with varying degrees of protection and access:

- **Zone A** – Full reserve
- **Zone B** – General reserve
- **Zone Bs** – Special general reserve
- **Zone C** – Partial reserve

In addition to its national designation, Capo Testa is also part of a Site of Community Importance (SCI ITB010007), incorporated into the EU Natura 2000 ecological network and subject to a dedicated management plan under the Habitats Directive (Regione Autonoma della Sardegna, 2025). This layered conservation status highlights the ecological relevance of the area and underscores the need for robust monitoring strategies.

The presence of clear waters, varying benthic substrates, and known meadows of *P.O.* make Capo Testa particularly well suited for satellite-based mapping approaches. The combination of ecological value, regulatory protection, and data availability was central to its selection for this study. Figure 3.2 shows the spatial extent of the Capo Testa MPA and the surrounding region used in the analysis.



ZONA A

Punto	Latitudine (°N)	Longitudine (°E)
A	41° 14′ 47.25″	009° 06′ 25.33″
B	41° 14′ 55.22″	009° 07′ 22.51″
C	41° 13′ 15.91″	009° 07′ 22.33″
D	41° 13′ 22.00″	009° 06′ 25.19″

ZONA B

Punto	Latitudine (°N)	Longitudine (°E)
N (in costa)	41° 15′ 11.42″	009° 12′ 47.77″
J	41° 15′ 32.25″	009° 12′ 47.84″
K	41° 16′ 03.75″	009° 13′ 59.16″
L	41° 14′ 56.89″	009° 16′ 13.05″
M (in costa)	41° 14′ 21.74″	009° 16′ 12.91″
G (in costa)	41° 13′ 00.25″	009° 09′ 48.83″
C	41° 13′ 15.91″	009° 07′ 22.33″
B	41° 14′ 55.22″	009° 07′ 22.51″
H	41° 15′ 18.62″	009° 10′ 10.87″
I (in costa)	41° 14′ 29.89″	009° 10′ 10.75″

ZONA Bs

Punto	Latitudine (°N)	Longitudine (°E)
I (in costa)	41° 14′ 29.89″	009° 10′ 10.75″
H	41° 15′ 18.62″	009° 10′ 10.87″
E	41° 15′ 28.98″	009° 11′ 25.57″
F (in costa)	41° 14′ 45.09″	009° 11′ 25.44″

ZONA C

Punto	Latitudine (°N)	Longitudine (°E)
T (in costa)	41° 11′ 52.27″	009° 09′ 13.52″
U	41° 13′ 12.45″	009° 05′ 57.98″
P	41° 15′ 25.96″	009° 05′ 58.18″
Q	41° 16′ 12.01″	009° 11′ 29.51″
R	41° 15′ 36.62″	009° 11′ 42.01″
S (in costa)	41° 14′ 33.06″	009° 11′ 41.91″
V (in costa)	41° 14′ 39.67″	009° 11′ 55.74″
W	41° 16′ 21.36″	009° 12′ 37.02″
X	41° 16′ 32.73″	009° 13′ 59.26″
Y	41° 15′ 12.04″	009° 16′ 40.92″
Z (in costa)	41° 13′ 49.79″	009° 16′ 40.57″

Figure 3.2: Zoning Map of Capo Testa Punta Falcone Marine Protected Area, Sardinia, Italy.(source: Area Marina Protetta Capo Testa Punta Falcone).

3.3 Satellite data and sources

Satellite remote sensing has become an indispensable tool in environmental monitoring, offering a broad range of data products that support applications in land use planning, agriculture, water quality assessment, disaster response, and coastal ecosystem analysis. The variety of satellite systems currently in operation includes both passive and active sensors, with spatial, spectral, and temporal resolutions tailored to specific scientific and operational needs. Platforms such as Landsat, MODIS, WorldView, and Sentinel offer users the ability to monitor Earth's surface consistently and repeatedly over time.

Among these, the Copernicus Earth Observation Program stands out as one of

the most comprehensive initiatives currently available. Developed and managed by the European Commission in partnership with the European Space Agency (ESA), Copernicus provides free and open access to satellite data from a growing constellation of Sentinel missions. Each Sentinel satellite series is designed for a particular type of observation: Sentinel-1 focuses on radar imaging, Sentinel-2 provides high-resolution optical imagery, Sentinel-3 collects ocean and land surface measurements, and other missions target atmospheric, climate, and emergency management applications.

The Sentinel-2 mission, which is central to this study, was launched in 2015 with the first satellite (Sentinel-2A), followed by Sentinel-2B in 2017. Together, these twin satellites orbit the Earth in a sun-synchronous configuration, allowing for global land coverage approximately every five days. The mission delivers high-resolution multispectral imagery suitable for vegetation, water bodies, and land cover mapping, and is especially well suited to coastal and marine studies due to its inclusion of water-penetrating spectral bands. Data can be accessed through the Copernicus Data Space Ecosystem (Copernicus Data Space Ecosystem, 2025).

Sentinel-2 data is available in two processing levels: Level-1C and Level-2A. Level-1C products contain top-of-atmosphere (TOA) reflectance values and require atmospheric correction to derive surface reflectance. In contrast, Level-2A products are already corrected for atmospheric effects and provide surface reflectance at the bottom of the atmosphere. Given the objectives of this research, which involve detecting subtle variations in underwater vegetation reflectance, only Level-2A products were used to ensure consistency and minimize preprocessing efforts (European Space Agency, 2025a).

Each downloaded Sentinel-2 Level-2A product contains data in three resolution folders: `R10m`, `R20m`, and `R60m`, corresponding to spatial resolutions of 10, 20, and 60 meters, respectively. These folders contain georeferenced image tiles of different spectral bands, which can be selectively used or resampled depending on the analytical requirements (ESA Sentinel Wiki, 2025).

The Sentinel-2 mission provides 13 spectral bands, each designed to capture specific reflectance features of the Earth's surface:

- **10m resolution:** B2 (Blue), B3 (Green), B4 (Red), B8 (NIR)
- **20m resolution:** B5, B6, B7 (Red Edge), B8A (Narrow NIR), B11, B12 (SWIR)
- **60m resolution:** B1 (Coastal aerosol), B9 (Water vapor), B10 (Cirrus) – B10 is not typically used in surface analysis

To achieve a uniform input resolution for our deep learning model, all 20 m and 60 m bands were resampled to 10 m resolution using bilinear interpolation. This approach

allows for a consistent pixel size across input features and simplifies the integration of multiple bands in the classification pipeline.

For the purposes of temporal analysis, this study focused on four distinct time epochs, selecting imagery from the month of September in the years 2015, 2018, 2021, and 2024. September was chosen to represent post summer conditions, where sea states are typically calm, water clarity is high, and vegetation is still active. This temporal spacing allows for the observation of long-term trends in the spatial extent and reflectance characteristics of *P.O.* meadows. The scene of each year was carefully selected to ensure minimal cloud cover, low turbidity, and suitable solar elevation angles (Figure 3.3).

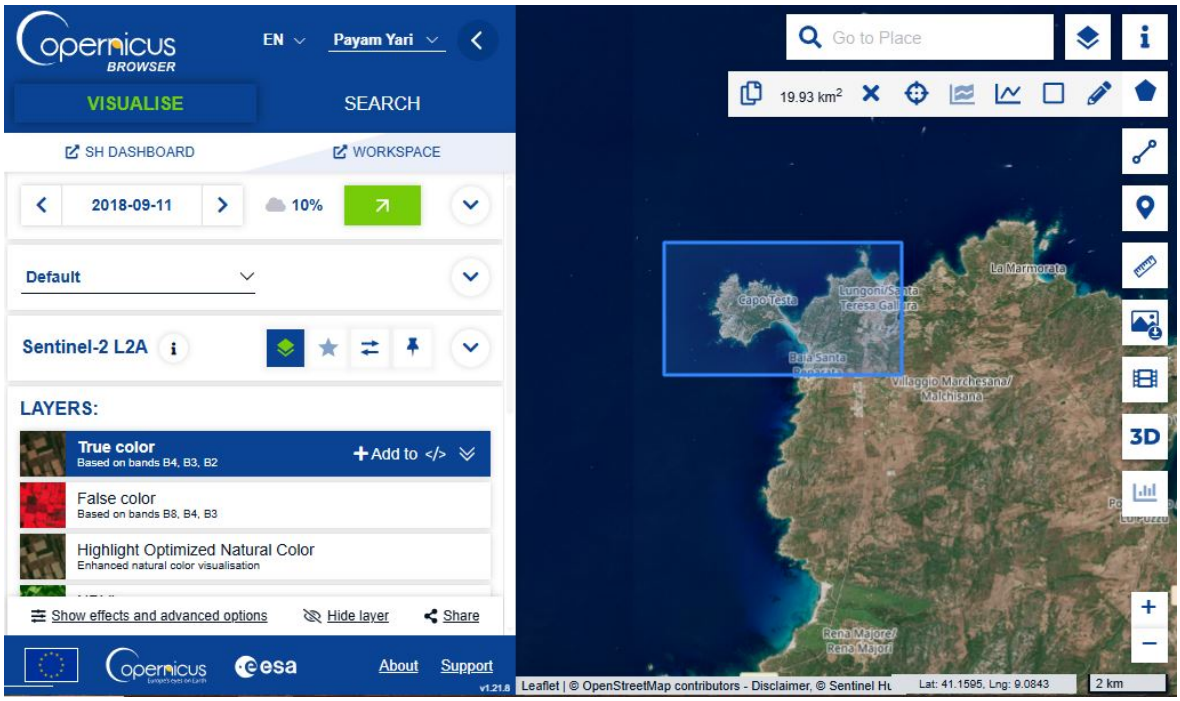


Figure 3.3: Copernicus Browser Visualise, September 2018

The availability of consistent and high-resolution Sentinel-2 imagery across these four temporal epochs offered a solid foundation for multitemporal classification of submerged vegetation. However, raw satellite data alone are not directly suitable for model training or analysis. It was therefore necessary to select specific bands for each year, resample all layers to a common spatial resolution, and spatially clip the scenes to match the study area boundaries. These steps were performed using ArcGIS Pro, where multiband composite images were created and exported for classification. The following section describes in detail the preprocessing procedures applied to the selected Sentinel-2 images prior to their integration into the deep learning workflow.

3.4 Preprocessing workflow

To prepare the satellite data for classification, a structured preprocessing pipeline was implemented using ArcGIS Pro. Each of the four temporal epochs was managed in a separate ArcGIS project to ensure consistency and clarity throughout the procedure. 3.4 illustrates the workflow for preparing data for deep learning training, covering the process from initial image preprocessing to the generation of training chips.

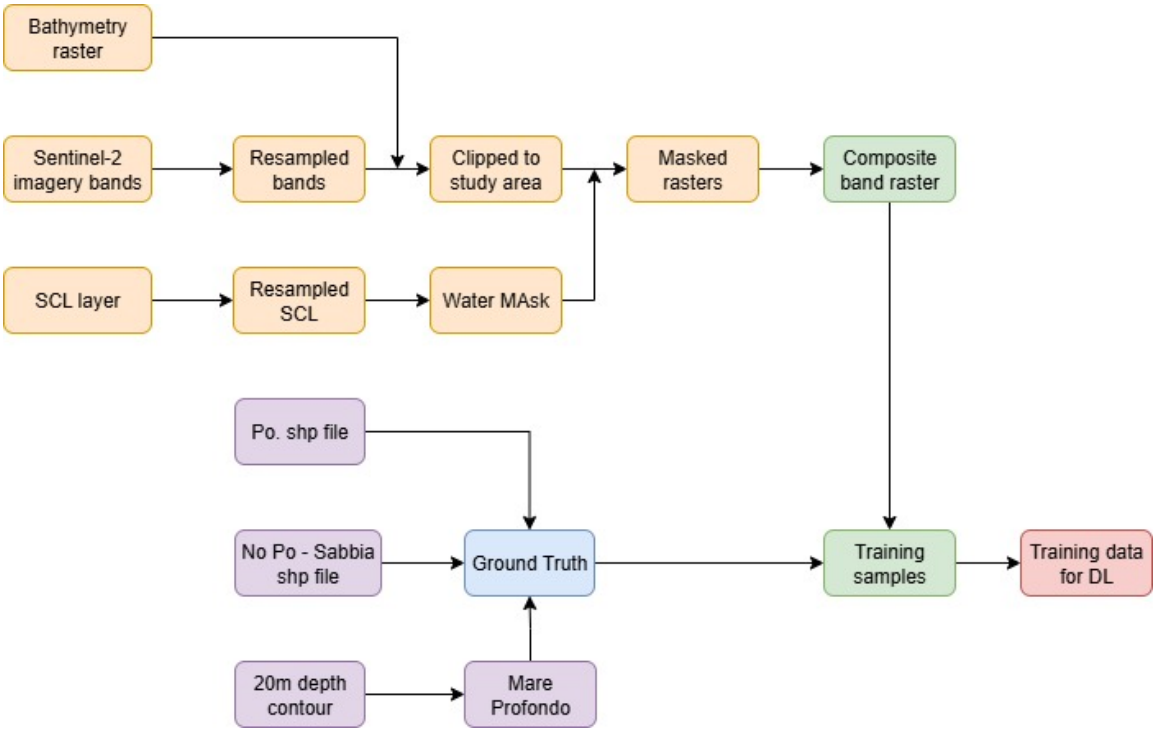


Figure 3.4: Diagram of preparing training data for deep learning.

For each year, the relevant Sentinel-2 Level-2A image was loaded into the project environment. To maximize spatial detail, all available bands from the three resolution levels were included: Bands 2, 3, 4, and 8 from the `R10m` folder; Bands 1, 5, 6, 7, 8A, 11, and 12 from the `R20m` folder; and Band 9 from the `R60m` folder. As all bands need to share a common resolution for multiband analysis, the rasters at 20 m and 60 m resolution were resampled to 10 m using ArcGIS Pro's "Resample" tool.

Among the available interpolation methods, the "Nearest" resampling technique was chosen. This method assigns the value of the nearest input cell to the output cell without altering the original pixel values, making it ideal for categorical or spectral classification tasks. Moreover, it preserves the spectral integrity of the bands, which is essential when analyzing vegetation reflectance in shallow marine environments (Esri, 2025a).

To manage file size and focus on ecologically relevant areas, a study area boundary was defined and applied across all rasters using the "Clip Raster" tool. This process

extracts only the portion of each band that intersects the area of interest, reducing data volume and ensuring all subsequent processing is spatially consistent.

In addition to spectral data, a bathymetry layer was incorporated to enhance the model's capacity to distinguish between submerged substrates. While traditional bathymetric mapping relies on acoustic systems such as single-beam or multi-beam echo sounders (SBES/MBES), these methods face limitations in shallow, dynamic coastal zones due to high operational costs and restricted accessibility. Airborne LiDAR, though useful, often suffers from similarly high costs and limited spatial coverage. As an alternative, satellite-derived bathymetry (SDB) techniques offer a rapid, low-cost means of estimating water depth across large areas by leveraging multispectral reflectance data. Recent studies have demonstrated the effectiveness of SDB using open Sentinel-2 imagery, especially when paired with additional altimetry or validation data (Bernardis et al., 2023).

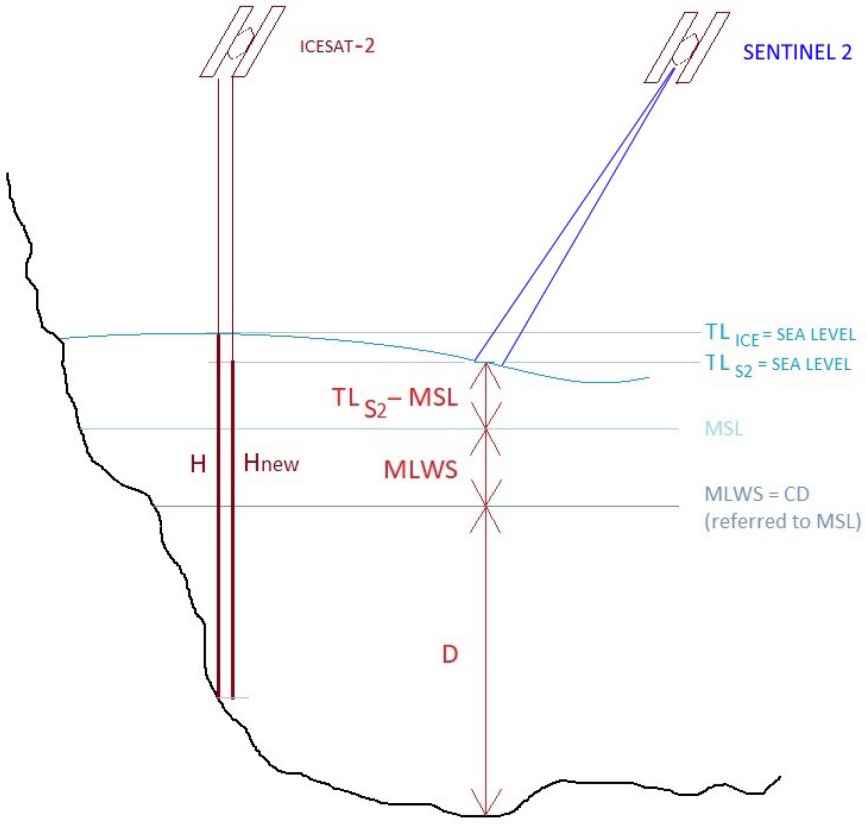


Figure 3.5: Schematic illustration of Sentinel-2 and ICESat-2 satellite altimetry reference levels used in satellite-derived bathymetry estimation. Adapted from (Bernardis et al., 2023).

In our study, a bathymetric raster was obtained from the Marine Geospatial Information Coastal Database (MaGIC) developed by the Italian Institute for Environmental Protection and Research (ISPRA) in collaboration with other national entities (ISPRA and partners, 2025). This dataset, which is part of a broader coastal mapping and management initiative, provides depth estimates for Italy's marine areas

derived from a combination of multibeam, singlebeam, and LIDAR surveys, integrated with other validated hydrographic sources. Incorporating this bathymetric layer alongside the spectral bands of Sentinel-2 enhances the model's ability to distinguish *P.O.* meadows from other benthic substrates that may have similar spectral responses but occur at different depths. Depth information is particularly valuable in complex coastal zones, where water column effects and bottom type variability can confound purely spectral classification.

Once all input rasters (12 spectral bands and 1 bathymetric band) were prepared, they were combined into a single composite image using the "Composite Bands" tool in ArcGIS Pro. The bands were added in the following order: B01, B11, B12, B02, B03, B04, B05, B06, B07, B08, B8A, B09, and Bathymetry. This consistent band stacking order was maintained across all years to ensure uniformity during model training and classification.

The resulting 13-band composite rasters, resampled to 10 m resolution, clipped to the study area, and standardized across four time periods, served as the final input for the deep learning classification workflow described in the next sections.

3.4.1 Water masking using the sentinel-2 scene classification layer

One of the key products of the Sentinel-2 Level-2A (L2A) dataset is the Scene Classification Layer (SCL), a thematic raster that assigns each pixel to a specific land cover or condition class. The SCL is generated at 20 m resolution by the Sen2Cor processor during atmospheric correction and accompanies each L2A product (Monks, 2023).

Table 3.1: Sentinel-2 Level-2A Scene Classification Layer (SCL) classes and pixel values (Monks, 2023).

Value	Class Description
0	No Data
1	Saturated or defective pixel
2	Dark area pixels
3	Cloud shadows
4	Vegetation
5	Not vegetated (bare soil, land)
6	Water
7	Unclassified
8	Cloud medium probability
9	Cloud high probability
10	Thin cirrus
11	Snow or ice

The SCL is designed to assist in cloud masking, land cover separation, and other pre-processing tasks. For marine habitat mapping, it is particularly useful for isolating water pixels from land and cloud-affected areas before classification. The layer includes class codes ranging from 0 to 11, each representing a different surface condition or object (Table 3.1).

In this study, the SCL raster was used to create a binary mask of water pixels (class value = 6). Since the SCL is distributed at 20 m resolution, it was resampled to 10 m using the nearest neighbour technique to preserve discrete class values and ensure alignment with the multispectral composite raster. The masking procedure for each acquisition date was as follows:

1. **Add the SCL raster:** Load the `*_SCL_20m.jp2` file for the selected Sentinel-2 scene into ArcGIS Pro.

2. **Resample to 10 m:** Apply the *Resample* tool (technique: Nearest) to match the spatial resolution of the composite raster.

3. **Create a binary mask:** Use the *Raster Calculator* to assign a value of 1 to water pixels (SCL = 6) and NoData to all others:

   ```
   Con("SCL_10m" == 6, 1)
   ```

4. **Apply the mask:** Multiply the composite raster by the binary mask, or use *Extract by Mask*, to retain only water-covered areas.

This procedure effectively removed land, cloud, and shadow pixels from the classification domain, ensuring that subsequent deep learning analysis focused exclusively on submerged regions.

3.4.2 Ground truth data

As part of the Poseidon project, researchers from the Department of Environment, Land and Infrastructure Engineering (DIATI) at Politecnico di Torino carried out a detailed mapping of seagrass meadows around Culuccia Island. In order to maintain methodological consistency and ensure that our results are comparable, we adopted the same classification scheme and general assumptions used in their work.

In that study, the ground truth dataset was organized into six distinct habitat classes:

1. Posidonia Degradata - Matte morta (Degraded Posidonia - dead matte)

2. Posidonia su matte-sabbia (Posidonia on matte-sand)

3. Posidonia su roccia (Posidonia on rock)

4. No Posidonia - Sabbia (No Posidonia - sand)

5. No Posidonia - Sedimenti vari (No Posidonia - mixed sediments)

6. Mare Profondo (Deep sea)

For our analysis, we prepared a ground truth shapefile covering the entire study area, ensuring that every location was assigned to one of these six classes. This was essential to avoid gaps or undefined regions, as any missing class assignments could compromise the training of the deep learning model.

The workflow for building this dataset began with defining the likely spatial extent of *P.O.*. Since the species is predominantly found in shallow waters, a polyline was drawn at the 20 m depth contour. Any areas located beyond this line, where no other habitat information was available, were assigned to the *Mare Profondo* (deep sea) class.

To define the Posidonia-related classes, we used an existing shapefile of *P.O.* meadows obtained from the European Marine Observation and Data Network (EMODnet) Seabed Habitats portal (European Marine Observation and Data Network (EMODnet), 2025). This dataset provides mapped distributions of *P.O.* across the Mediterranean, compiled from multiple national and regional monitoring programmes. From this dataset, the three relevant classes were extracted: *Posidonia Degradata-Matte morta*, *Posidonia su matte-sabbia*, and *Posidonia su roccia*.

Additional data were obtained from the National Research Council of Italy's (CNR) GeoNetwork portal (Consiglio Nazionale delle Ricerche (CNR), 2025), specifically from the record available at: `http://libeccio.bo.ismar.cnr.it:8080/geonetwork/pnrr/api/records/115e1ddb-0fc5-4b09-a437-ba64f5d0eaff`. From this dataset, the *No Posidonia - Sabbia* class was extracted.

Finally, the remaining areas within the study boundary that were not covered by any of the above layers were classified as *No Posidonia - Sedimenti vari*. This step ensured that the final ground truth shapefile represented a complete and seamless classification of the study area, ready for use in training and validating the classification model. The final ground truth layer, covering the entire study area without gaps, is shown in Figure 3.6.

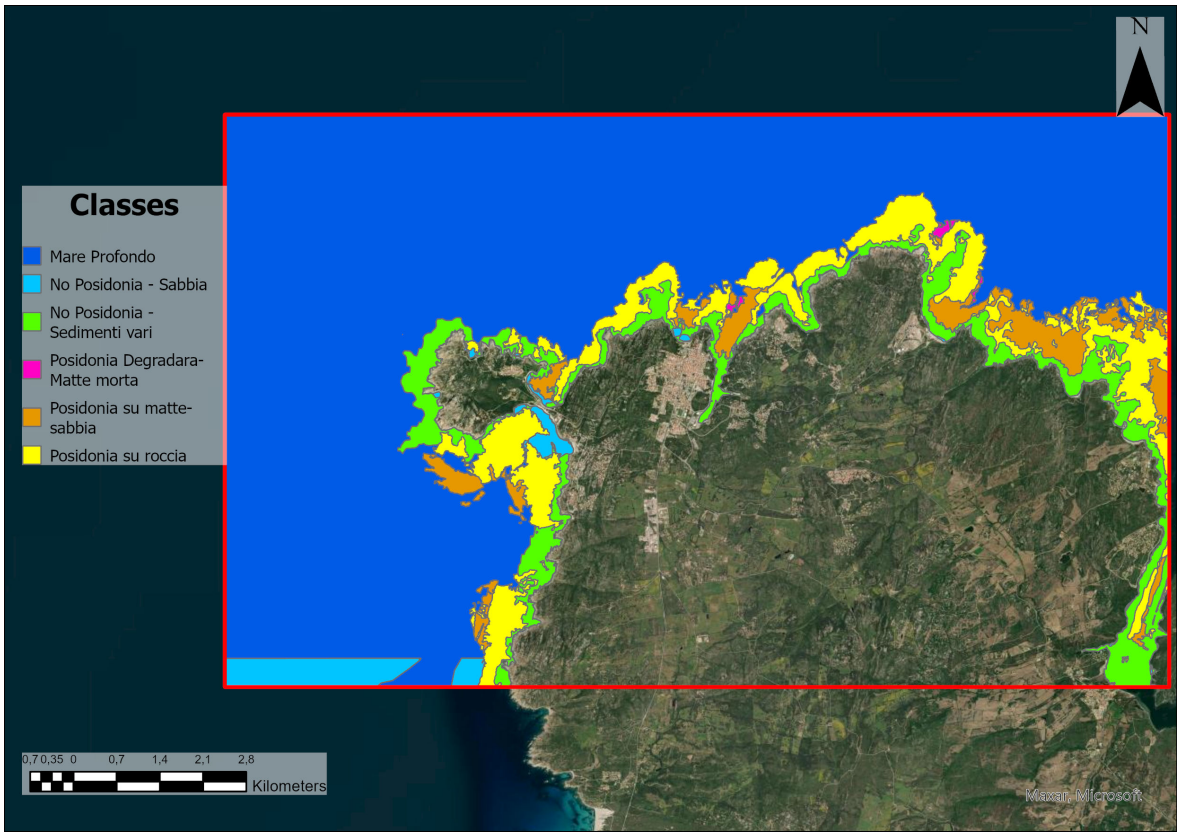


Figure 3.6: Ground truth habitat classification for the study area, based on six classe

3.4.3 Training sample definition

With the ground truth shapefile completed and the final 13-band composite raster prepared (Section 3.4), the next step was to create the training dataset required for deep learning classification. This was carried out in ArcGIS Pro using the *Training Samples Manager* tool, accessible from the *Imagery* tab under *Classification Tools*.

The Training Samples Manager provides an interactive environment for creating, editing, and managing training samples, which are pixel-based representations of known land cover or habitat classes. These samples form the reference dataset used to train supervised classification models, including deep learning approaches (Esri, 2025b). The tool allows the user to:

- Digitize polygons or rectangles directly on the imagery.
- Assign each sample to a predefined class from a user-defined schema.
- Review and edit samples to ensure accuracy and remove errors.
- Track the number of samples collected per class.

For this study, the six habitat classes defined in Section 3.5 were adopted: *Posidonia degradata – matte morta*, *Posidonia su matte sabbia*, *Posidonia su roccia*, *No Posidonia*

– *sabbia*, *No Posidonia – sedimenti vari*, and *Mare profondo*. Using the composite raster as the base image and the ground truth layer as a spatial guide, multiple training polygons were collected for each class.

To maximize the representativeness of the training set, sample collection followed these principles:

- Ensure coverage of the full spectral variability within each class, including differences due to water depth, substrate type, and lighting.
- Avoid boundary areas between classes to reduce the likelihood of mixed pixels.
- Distribute samples across the full spatial extent of the study area.

Once the desired number of samples was obtained, the Training Samples Manager exported them as a labeled raster dataset, with each pixel assigned to its respective class code. This labeled raster formed the core input for the deep learning model training described in Chapter 4.

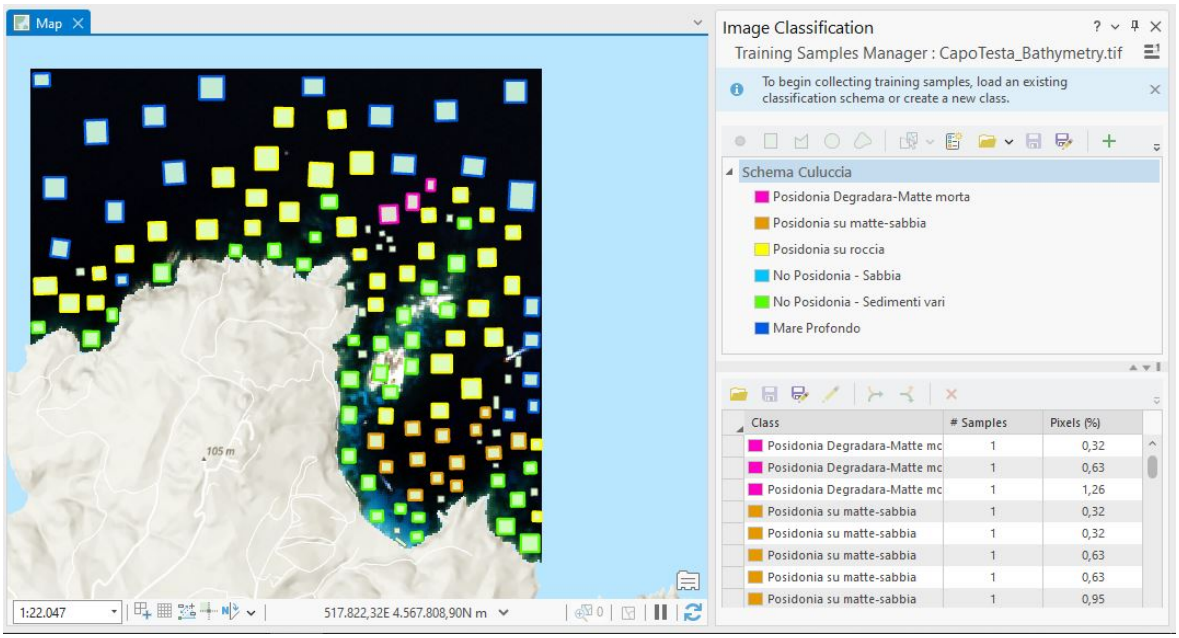


Figure 3.7: Example of training samples created in ArcGIS Pro's Training Samples Manager for the six habitat classes.

Chapter 4: Deep Learning

4.1 Introduction: from artificial intelligence to deep learning

Artificial Intelligence (AI) is a broad domain of computer science focused on building systems capable of tasks that traditionally require human intelligence, such as language processing, visual recognition, autonomous decision-making, and prediction. AI is now embedded in everyday technologies including virtual assistants, recommendation systems, medical diagnostics, and climate modeling. In environmental sciences, AI has emerged as a powerful tool for processing the vast amounts of data generated by remote sensing and monitoring systems.

Within AI, two subfields have gained particular importance: **Machine Learning (ML)** and **Deep Learning (DL)**. Machine learning refers to algorithms that learn patterns from data and improve their performance over time, often relying on hand-crafted features designed by experts. Deep learning, by contrast, employs multi-layered neural networks that can automatically learn hierarchical features directly from raw data, making it particularly effective for high-dimensional and complex inputs such as images.

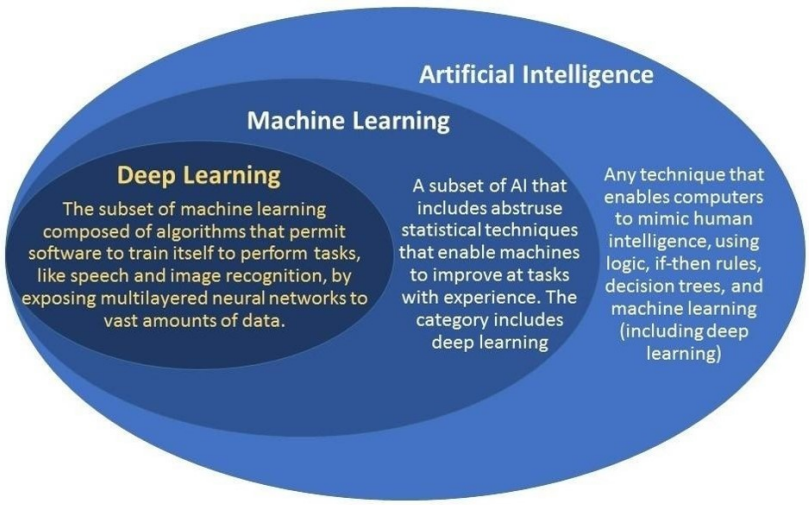


Figure 4.1: Conceptual relationship between Artificial Intelligence, Machine Learning, and Deep Learning.

Figure 4.1 illustrates the hierarchical relationship between these concepts. AI forms the broadest category, encompassing all computational approaches designed to emulate aspects of intelligence. ML is a subset of AI focused on data-driven learning. DL is a further subset of ML, distinguished by the use of neural networks with many layers that extract features and perform predictions in an integrated manner.
A closely related and rapidly emerging field is **GeoAI**, which represents the intersection of geospatial sciences, artificial intelligence, and remote sensing (Figure 4.2). GeoAI combines spatial science, AI (especially machine learning and deep learning), data mining, and high-performance computing to extract knowledge from large geospatial datasets. These technologies provide new opportunities for handling and analyzing spatial data more efficiently, improving accuracy in environmental monitoring, and developing integrative frameworks that connect GIS with other scientific domains (Alastal et al., 2022).

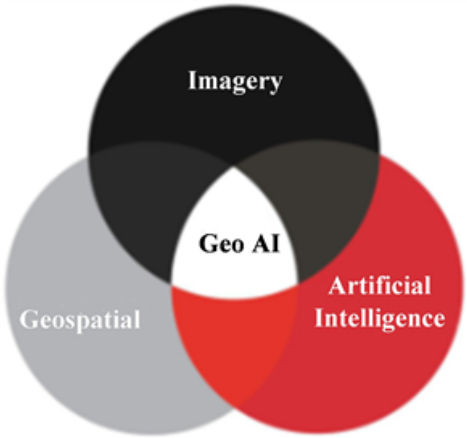


Figure 4.2: Conceptual framework of GeoAI (Alastal et al., 2022).

In the context of this thesis, DL is used for **semantic segmentation**, the pixel-wise classification of satellite images. This is crucial for mapping *P.O.* meadows, where spectral similarity to sand or other substrates makes traditional classification approaches insufficient. Using DL architectures such as U-Net and DeepLabv3 within GIS environments, We can obtain relatively accurate spatial maps of seagrass meadows that support both ecological monitoring and coastal engineering applications.

4.2 Deep learning fundamentals

4.2.1 Historical background

Deep learning (DL) emerged from the study of artificial neural networks (ANNs), which were originally inspired by simplified models of the brain's information processing (Ramachandran et al., 2021). Early milestones include McCulloch and Pitts' formal

neuron model (1940s) and Rosenblatt's *Perceptron* (1950s), one of the first trainable learning machines (Tappert, 2019). Later, the use of error backpropagation enabled multi-layer networks, but practical progress accelerated only when large labeled datasets and fast computation became available. A turning point came with *AlexNet*, a deep convolutional neural network (CNN) that dramatically improved ImageNet performance and demonstrated the practical power of deep models for large-scale image recognition (Krizhevsky et al., 2017). Since then, better software, GPUs/accelerators, and systematic engineering have made DL the dominant approach for many perception tasks (Watson et al., 2022; Armeniakos et al., 2022).

4.2.2 Artificial neural networks (ANNs)

An ANN is a stack of layers composed of units (neurons). A neuron takes a weighted sum of its inputs, adds a bias, and applies a nonlinear activation (e.g., ReLU). Layer by layer, the network transforms raw inputs into progressively more abstract features. Training proceeds by:

1. **Forward pass:** compute predictions given current weights.
2. **Loss:** measure the discrepancy between predictions and targets.
3. **Backward pass:** compute gradients by backpropagation.
4. **Update:** adjust weights with an optimizer (e.g., SGD, Adam).

This end-to-end learning features from data, distinguishes DL from pipelines that rely on manual feature engineering (Pakela et al., 2022; Ramachandran et al., 2021).

4.2.3 Convolutional neural networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for image and spatial data. Unlike traditional fully connected neural networks, CNNs exploit the spatial structure of images through localized connections and weight sharing, which reduces the number of parameters and makes them more efficient for high-dimensional inputs (Watson et al., 2022; Pakela et al., 2022). A CNN is typically composed of several key layers:

- **Convolutional layers:** apply small filters (kernels) to extract local features such as edges, corners, or textures.
- **Activation functions (e.g., ReLU):** introduce non-linearity to enable the learning of complex patterns.

- **Pooling layers:** downsample feature maps, reducing their spatial size while retaining important information.
- **Fully connected layers:** combine extracted features to form a global representation, usually for classification tasks.

Through multiple stacked convolutional and pooling layers, CNNs progressively transform raw input images into high-level feature representations (Figure 4.3). This hierarchical learning enables the network to first detect low-level features (edges, colors), then intermediate features (shapes, textures), and finally high-level concepts (objects or regions of interest). Such properties have made CNNs the standard in modern computer vision tasks such as classification, object detection, and semantic segmentation (Krizhevsky et al., 2017; Watson et al., 2022).

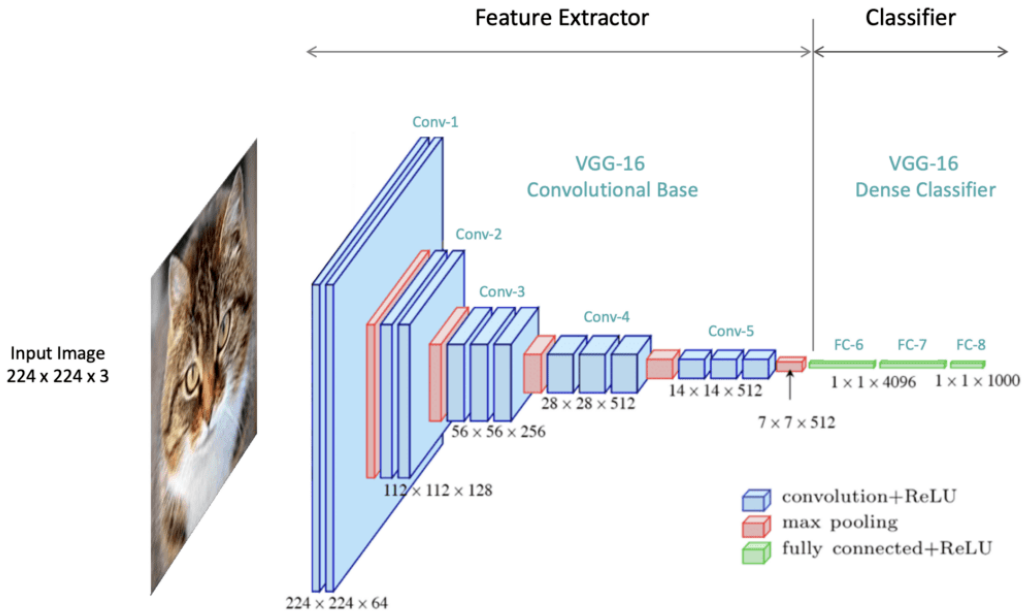


Figure 4.3: Schematic of a CNN (Mallick, 2016).

4.2.4 Kernels, stride, and padding in CNNs

The fundamental operation in CNNs is the **convolution**, where a small matrix of numbers (called a kernel or filter) slides across the input image. At each position, the kernel values are multiplied by the underlying pixel values and summed to produce a new value in the output feature map. This process enables the network to highlight specific visual characteristics in the image, such as edges or textures (Jain, 2019).

Different kernels extract different features. For example, one kernel may blur the image, another may sharpen it, and another may detect edges. During training, the network learns the optimal kernel weights for the task at hand. Figure 4.4 shows examples of commonly used kernels.

Original	Gaussian Blur	Sharpen	Edge Detection
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
			

Figure 4.4: Examples of different kernels and their effects on an image, including Gaussian blur, sharpen, and edge detection. Source: adapted from (Jain, 2019).

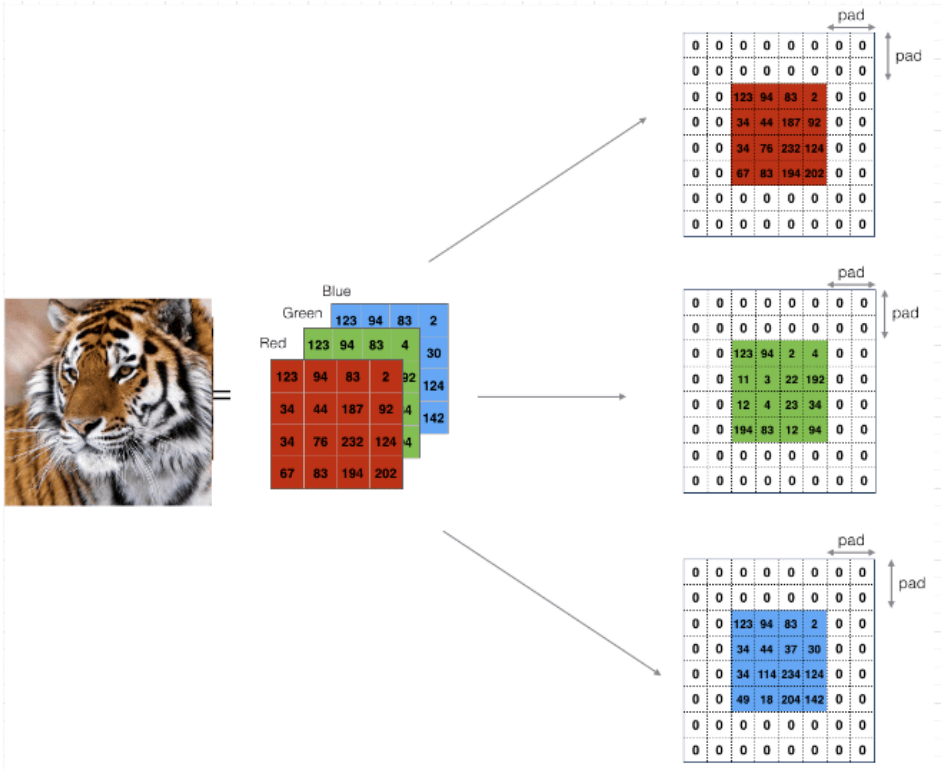


Figure 4.5: Illustration of padding in convolution. Source: adapted from (Mallick, 2016).

Two additional parameters control the convolution process:

- **Stride:** determines the number of pixels the kernel moves at each step. A stride of 1 results in high-resolution feature maps; larger strides reduce spatial resolution and computation.
- **Padding:** adds artificial pixels (often zeros) around the borders of the input to preserve spatial dimensions and avoid information loss at the edges (Figure 4.5).

An example of the convolution operation is shown in Figure 4.6. Here, a 3×3 kernel is applied to a region of the source image, producing a single output value in the destination layer.

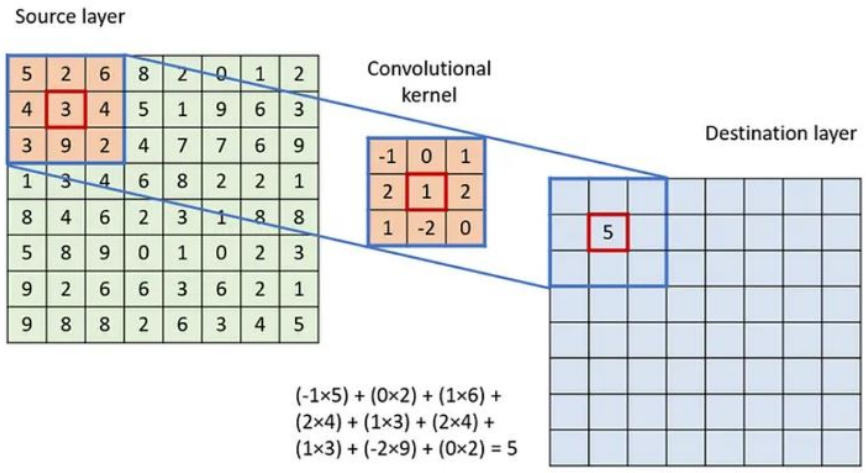


Figure 4.6: Example of a convolution operation using a 3×3 kernel applied to a source image region. Source: adapted from (Mallick, 2016).

Finally, **pooling layers** (such as max pooling) are often applied after convolution to downsample the feature maps. Pooling reduces computation, makes the model more efficient, and adds robustness by retaining only the most significant responses. Together, convolution, activation, padding, stride, and pooling enable CNNs to learn complex visual patterns in a structured and efficient way.

4.3 U-Net architecture

4.3.1 Architectures in deep learning

In deep learning, an **architecture** defines the arrangement of layers and the flow of information in a model. Different architectures are optimized for different tasks: for example, classification networks (e.g., VGG, ResNet) map an image to a single class label, while segmentation architectures predict a label for each pixel. In this context, U-Net represents one of the most widely used encoder–decoder architectures for semantic segmentation, originally proposed for biomedical imaging (Ronneberger et al., 2015b) and later adapted for remote sensing and satellite imagery (Anonymous, 2020).

4.3.2 Encoder–decoder design

The U-Net follows a symmetric encoder–decoder structure:

- **Encoder (contracting path):** progressively reduces spatial dimensions using repeated convolutional layers and pooling, capturing increasingly abstract features. This is similar to a standard CNN feature extractor.

- **Decoder (expansive path):** progressively upsamples feature maps using transpose convolutions (up-convs) to restore spatial resolution and generate dense predictions at the pixel level.
- **Skip connections:** a key innovation of U-Net is the direct transfer of feature maps from the encoder to the decoder at corresponding scales. These skip connections preserve fine-grained spatial information that would otherwise be lost during downsampling, significantly improving localization accuracy.

Figure 4.7 shows a U-Net architecture, characterized by a symmetric encoder–decoder structure with skip connections linking corresponding layers.

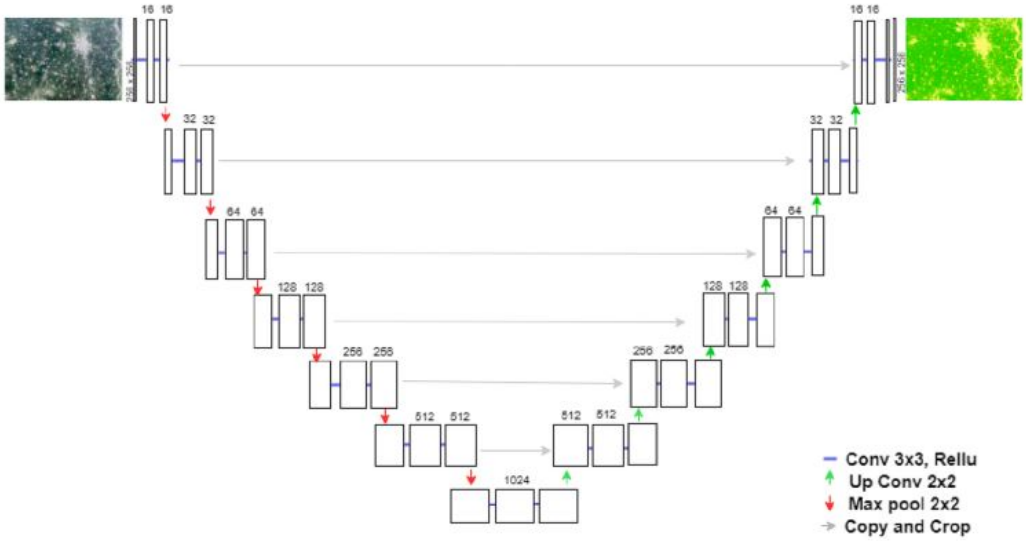


Figure 4.7: Architecture of the U-Net model for image segmentation. Adapted from (Ronneberger et al., 2015b).

4.3.3 Role of U-Net in semantic segmentation

Unlike image classification, semantic segmentation requires assigning a class to every pixel in the image. The encoder captures global context, while the decoder reconstructs fine detail. By combining low-level features (edges, textures) with high-level semantic features (object identity), U-Net enables accurate delineation of objects, even in cases of complex boundaries (Ronneberger et al., 2015b; Esri Developer, 2023; GeeksforGeeks, 2023b).

4.3.4 Variants and backbones

Since its introduction, U-Net has inspired numerous variants:

- **ResU-Net:** integrates residual connections (from ResNet) to improve gradient flow and training stability.

- **Dense U-Net:** incorporates dense connectivity (from DenseNet) to promote feature reuse.
- **Efficient U-Net:** uses EfficientNet as a backbone to balance accuracy and efficiency.

In practice, the encoder can be replaced by different pre-trained classification networks (e.g., ResNet, EfficientNet) to leverage transfer learning. In our work, we use a U-Net with a DeepLabv3 backbone for semantic segmentation of *P.O.*.

4.4 Semantic segmentation and backbones

4.4.1 Semantic segmentation

Semantic segmentation is the process of assigning a class label to every pixel in an image, effectively producing a dense classification map. This is distinct from image classification, which assigns a single label to the entire image, and object detection, which localizes objects with bounding boxes. Semantic segmentation is particularly important in environmental remote sensing, where it enables the delineation of habitats, land cover types, or vegetation patches at a fine spatial scale.

For this thesis, semantic segmentation is essential for accurately mapping *P.O.* meadows, whose reflectance often overlaps with sandy or rocky substrates. Traditional classification approaches struggle with this spectral similarity, while semantic segmentation models can exploit both local texture and contextual information to produce precise pixel-wise maps.

4.4.2 DeepLabv3 architecture

One of the most influential semantic segmentation architectures is **DeepLabv3**, proposed by Chen et al. (2017b). Its central innovation is the use of **atrous (dilated) convolutions**, which allow convolutional filters to capture larger context without reducing the resolution of feature maps. This approach improves the model's ability to balance fine spatial detail with global context.

DeepLabv3 also integrates **Atrous Spatial Pyramid Pooling (ASPP)**, which applies atrous convolutions with multiple dilation rates in parallel. This design allows the network to capture features at multiple scales, a crucial advantage for images where objects vary in size and shape (Datature, 2023; Berríos, 2020). By combining local and global information, DeepLabv3 achieves high accuracy in segmentation benchmarks and has become a standard in many applied fields, including medical imaging, autonomous driving, and remote sensing.

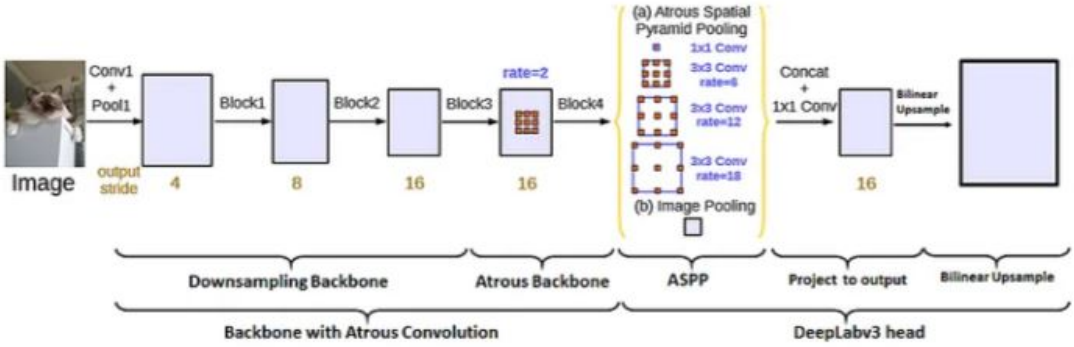


Figure 4.8: Architecture of DeepLabv3, Adapted from (Chen et al., 2017b).

Figure 4.8 illustrates the use of atrous convolutions and Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale contextual features.

4.4.3 Backbones in semantic segmentation

DeepLabv3, like other modern segmentation models, separates its design into two components:

- **Architecture:** the overall framework (e.g., DeepLabv3, U-Net), which defines how features are extracted, processed, and upsampled for pixel-level predictions.
- **Backbone:** the feature extractor used as the encoder. Typically, well-established image classification networks such as ResNet, Xception, or EfficientNet serve this role, providing powerful hierarchical representations of input images.

Among the different backbones available, **Residual Networks (ResNet)** have become one of the most widely adopted in semantic segmentation models. ResNet introduces **residual connections**, which allow the input of a layer to bypass one or more intermediate layers and be added directly to the output. This design addresses the problem of vanishing gradients, enabling the training of very deep networks without performance degradation.

Different ResNet variants (ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152) vary in depth and complexity, trading off between computational cost and feature representation power. Shallower networks (e.g. ResNet-18, ResNet-34) are faster and less memory intensive, while deeper networks (e.g. ResNet-101, ResNet-152) capture richer and more abstract features but require more resources (Code, 2023).

In this thesis, ResNet backbones were selected due to their proven robustness, availability of pre-trained weights, and compatibility with architectures such as DeepLabv3 and U-Net. This modularity ensures flexibility: the *architecture* (e.g. DeepLabv3) defines how segmentation is performed, while the *backbone* (e.g. ResNet-50) extracts the essential multiscale features from raw imagery.

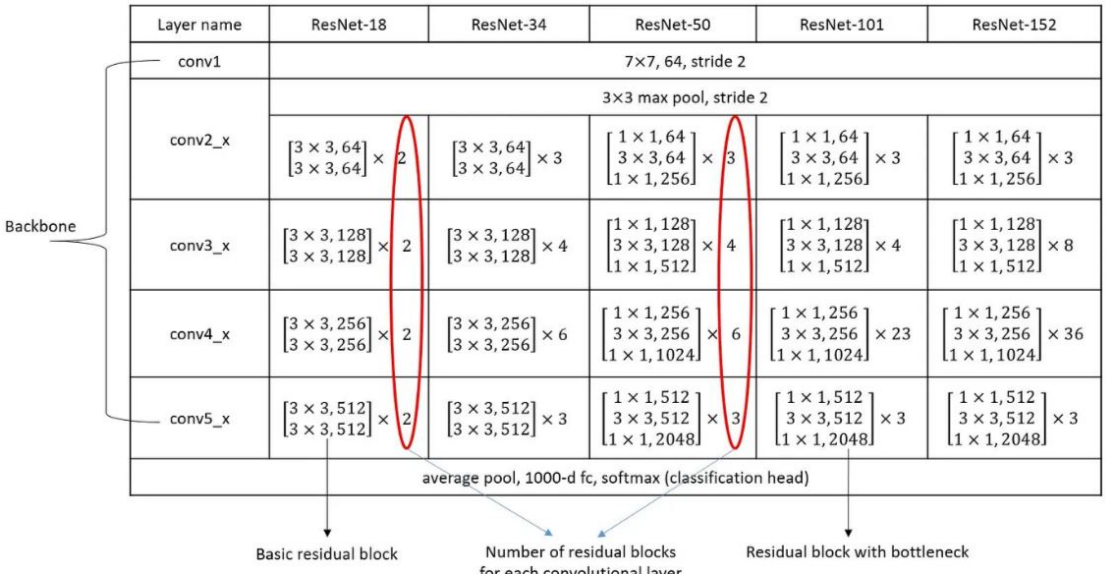


Figure 4.9: Comparison of ResNet architectures of varying depths, showing the number and type of residual blocks in each stage. Source: adapted from ResNet tutorial (Code, 2023).

As illustrated in Figure 4.9, deeper ResNet variants provide more powerful feature extraction at the cost of higher computational demand. Table 4.1 summarizes the main characteristics of the commonly used ResNet backbones. This comparison highlights the trade-off between model complexity and performance and explains why intermediate variants (e.g. ResNet-50) are often selected as a balance between accuracy and efficiency in semantic segmentation tasks.

Table 4.1: Comparison of commonly used ResNet backbone variants. Deeper models capture richer features but require more computational resources.

Backbone	Layers	Parameters (M)	Advantages	Limitations
ResNet-18	18	∼11M	Fast, lightweight	Limited feature depth
ResNet-34	34	∼21M	Good balance of speed/accuracy	Less powerful than deeper models
ResNet-50	50	∼25M	Strong feature extraction, widely used	Higher computational cost
ResNet-101	101	∼44M	Captures very rich features	Slower training and inference
ResNet-152	152	∼60M	Very deep representation capacity	Computationally expensive

The column "Parameters (M)" indicates the approximate number of trainable parameters in millions. Models with more parameters have higher representational capacity, but also demand greater computational resources and training data.

4.5 Exporting training data for deep learning

After defining the training samples, the next step in our workflow was to use the *Export Training Data for Deep Learning* tool in ArcGIS Pro. This tool prepares image chips and their corresponding labels in a format suitable for deep learning frameworks such as PyTorch or TensorFlow. Since deep learning models cannot be

trained directly on full-resolution satellite images, the tool automatically slices the input raster and annotation layers into smaller *tiles*, ensuring that the training dataset is both manageable and consistent.

The tool requires several key parameters:

- **Tile Size (X, Y):** defines the width and height of each extracted tile. We tested tile sizes of 10, 20, 30, and 40 pixels. A size of **30 × 30** produced the most accurate results, as it provided sufficient spatial context without excessively reducing the number of training samples.
- **Stride (X, Y):** determines the degree of overlap between adjacent tiles. To ensure boundary regions were adequately captured, we set the stride equal to half of the tile size (e.g. stride = 15 when tile size = 30).
- **Metadata Format:** specifies how training labels are stored. We selected **Classified Tiles**, in which each image tile is paired with a corresponding labeled tile. This format is specifically suited for semantic segmentation tasks.

The combination of tile size and stride is particularly critical: smaller tiles (10, 20 pixels) captured fine detail but lacked larger spatial context, while larger tiles (40 pixels) increased computational cost and reduced the diversity of training patches. The choice of **30 × 30 with stride = 15** achieved the best balance between capturing detail and providing context, leading to improved segmentation performance.

4.6 Training deep learning models

After exporting the chips (Section 4.5), model training was organised as a sequence of controlled experiments rather than a single run. We began with the 2015 scene and first trained on the full study area shown in Fig. 3.6. This gave many samples but led to very long runtimes and strong spectral variability across the bay. Depth, turbidity, and substrate change between sectors, so the same class can look different from place to place. After several trials we restricted the domain to a smaller rectangle that still included all six classes (Fig. 4.10). This sub-area reduced training time, improved consistency, and made it easier to isolate the effect of each design choice.

To study the impact of sampling, we prepared three training-sample sets: (i) many small polygons, (ii) fewer but larger polygons, and (iii) a mixed set combining both. For each sample set we exported four datasets that differed only in chip size (10, 20, 30, and 40 pixels); all other inputs were kept identical (band stack, bathymetry layer and masks). This design let us test one factor at a time while controlling the rest.

Model testing focused on two semantic segmentation families, *DeepLabv3* and *U-Net*, with ResNet encoders (mainly ResNet-34 and ResNet-50). In each run we

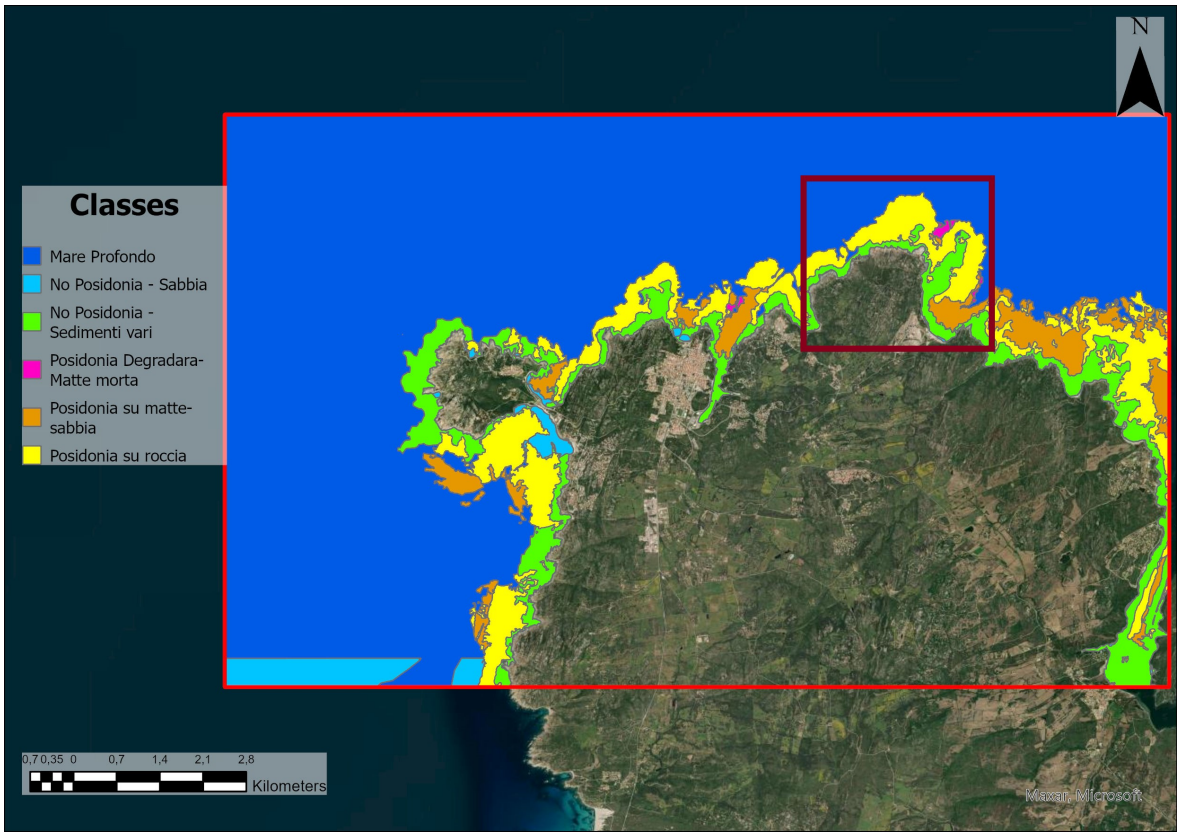


Figure 4.10: Sub-area used for controlled training experiments. The rectangle is drawn on the ground truth map represented in Fig. 3.6.

changed a single element (chip size, validation split, class weighting, or oversampling) and kept the remaining settings fixed (batch size, band order, preprocessing).

Table 4.2: An example of Recorded Metrics for 2015 dataset classification

Test	Architecture	Backbone	Tile	Stride	W1	W2	W3	W4	W5	W6	Padding	Accuracy	Kappa
1	U-net	ResNet50	10	5	60	10	9	4	5	1	4	0.75	0.67
				F1-Score	0	0.104	0.055	0.397	0.204	0.696			
2	U-net	ResNet50	10	5	500	150	2	60	50	50	4	0.701	0.61
				F1-Score	0	0.205	0.484	0.727	0.502	0.293			
3	DeepLabv3	ResNet34	10	5	500	150	2	60	50	50	4	0.717	0.632
				F1-Score	0.072	0.342	0.454	0.558	0.489	0.287	1	0.758	0.678
4	U-net	ResNet50	10	5	4.8152	1.9418	0.4318	34.0506	0.7893	0.6006	4	0.69	0.61
				F1-Score	0.053	0.342	0.357	0.727	0.62	0.327	1	0.742	0.665
5	DeepLabv3	ResNet34	10	5	4.8152	1.9418	0.4318	34.0506	0.7893	0.6006	4	0.739	0.662
				F1-Score	0.094	0.354	0.354	0.6	0.593	0.394	2	0.733	0.655
											1	0.752	0.679
6	DeepLabv3	ResNet34	30	15	40.8152	1.9418	0.4318	34.0506	0.7893	6.006	4	0.825	0.77
				F1-Score	0	0.439	0.431	0.75	0.612	0.402	7	0.815	0.758
											14	0.819	0.763
7	DeepLabv3	ResNet34	30	15	40.8152	4.9418	0.4318	30.0506	0.7893	6.006	2	0.812	0.754
				F1-Score	0.411	0.449	0.341	0.644	0.633	0.178	3	0.815	0.758
											14	0.819	0.762
8	DeepLabv3	ResNet34	30	15	42.8152	5.9418	0.5318	28.0506	0.7893	7.006	2	0.847	0.797
				F1-Score	0	0.287	0.361	0.75	0.664	0.346	7	0.85	0.802
											14	0.866	0.823
9	DeepLabv3	ResNet34	30	15	40.8152	5.9418	1.4318	28.0506	0.8893	8.006	2	0.847	0.797
				F1-Score	0	0.4179	0.3816	0.4615	0.6574	0.318	7	0.85	0.802
											14	0.866	0.823
10	DeepLabv3	ResNet34	30	15	40.8152	4.9418	0.4318	30.0506	0.7893	6.006	4	0.8825	0.8433
				F1-Score	0.0982	0.4383	0.5937	0.7	0.8066	0.4529	7	0.8539	0.8045
											14	0.8793	0.8386

4.6.1 Class weights and oversampling

To keep the training focused on *Posidonia* habitats, we balanced the loss with class weights and combined this with selective oversampling. The initial weights were computed automatically by the script provided in the Appendix: the code counts the pixels of each class and assigns weights from their frequency (rarer classes receive larger weights). We treated these as a first guess and then adjusted them per dataset (2015, 2018, 2021, 2024) after short pilot runs, because both spatial patterns and class proportions change across years and the weights have a strong impact on the outcome.

Across all years, the three seagrass classes (1–3) were given higher weights than the others, and *Posidonia degradata - matte morta* (class 1) consistently received the largest weight. This class is rarer and more fragmented; without extra emphasis the model tended to underfit it. In parallel, we oversampled the seagrass classes during mini-batch construction so that they appeared more often during training.

We also ran sensitivity checks while keeping the rest of the settings fixed. Examples include setting the weight of one class temporarily to 1 while leaving the others unchanged, oversampling only class 1, and in other tests using oversampling classes 1–5. These tests were useful to understand interactions between weighting and sampling, but they did not yield a single trend that held in every year. The final choice of weights and oversampled classes was therefore made per dataset, guided by validation Dice and visual checks of the predicted maps.

4.6.2 Chip size and validation split

The *chip size* parameter in training must be compatible with the image tiles exported in Section 4.5. To evaluate the effect of context size on model performance, we tested chip sizes of 10, 20, 30, and 40 pixels. Results indicated that a size of **30 pixels** provided the best balance between detail and context, and this configuration was adopted for the final models.

The *validation split* determines the fraction of training data reserved for validation. We tested splits of 20%, 25%, and 30%. A split of **20%** was selected as the optimal setting, ensuring a sufficiently large validation set while retaining most data for training.

4.6.3 Learning rate scheduling

The learning rate sets the step size of each update. A fixed value can be too small (slow) or too large (unstable), so we used a multi-phase decay: start higher to capture coarse structure, then reduce it to refine the solution. All runs followed the schedule in Table 4.3. This gave fast early convergence and smoother validation curves, and kept the protocol consistent so that differences across tests could be linked to other factors.

Table 4.3: Learning rate schedule used during training.

Epochs	1400	1100	800	600	400
Learning rate	4.0×10^{-4}	3.0×10^{-4}	2.0×10^{-4}	1.0×10^{-4}	5.0×10^{-5}

4.6.4 Monitored metrics and optimization

We tracked four indicators during training: overall accuracy, Dice (F1) score, training loss, and validation loss. After each phase of the learning-rate schedule, we reviewed the curves and saved the best checkpoint based on validation Dice, supported by a quick visual check of the predicted maps. These signals guided small, targeted changes to the hyperparameters (e.g., class weights, oversampling choice, chip size and validation split) so that we improved generalization without inflating runtime.

The configuration that proved most reliable on the 2015 pilot was then used as a starting point for 2018, 2021, and 2024, with light adjustments only when the validation metrics or map inspection suggested a mismatch. The Python script used for training and logging these metrics is included in the Appendix.

Chapter 5: Classification and Results

5.1 Introduction

This chapter presents the results of the classification of *P.O.* across the study area for the four reference years: 2015, 2018, 2021, and 2024. The outputs are reported in a structured way, following the sequence of the workflow described in the previous chapter.

The chapter first introduces the spectral signatures extracted for Capo Testa Island, which illustrate the reflectance patterns of *Posidonia* compared with other benthic classes across different spectral bands. These signatures provided a first indication of the separability of classes and helped to interpret the subsequent classification results.

The next part focuses on the classification process itself. Each yearly raster was generated using the *Classify Pixels Using Deep Learning* tool in ArcGIS Pro, which applied the trained deep learning package (.dlpk). Specific technical considerations, such as the role of the padding parameter, are discussed since they affect how image tiles are processed at the borders. The training process also produced diagnostic plots, including training loss, validation loss, accuracy, and F1 score, which are reported in a 2×2 figure layout for each year.

The accuracy of the classification outputs was then assessed within ArcGIS Pro using three tools from the Image Analyst toolbox: *Create Accuracy Assessment Points*, *Update Accuracy Assessment Points*, and *Compute Confusion Matrix*. The resulting confusion matrix provided detailed insight into model performance by reporting correctly and incorrectly classified pixels for each class. From this matrix, key evaluation indices such as User's Accuracy, Producer's Accuracy, and the Kappa coefficient were derived.

Finally, a multi-temporal analysis was conducted to compare the classification results across different years. This allowed the detection of spatial changes in the extent of *P.O.* meadows, identifying areas of possible regression or expansion and providing the basis for change detection analysis.

5.2 Spectral signatures

The spectral profiles for Capo Testa Island were generated from the composite raster using the *Spectral Profile Chart* tool in ArcGIS Pro (ESRI, 2024). This tool extracts pixel values across multiple bands for user-defined training samples and displays them as curves, providing an effective way to visualize the separability of classes.

Figure 5.1 shows the spectral signatures obtained for the six defined classes: (1) *Posidonia Degradata - Matte morta*, (2) *Posidonia su matte-sabbia*, (3) *Posidonia su roccia*, (4) *No Posidonia - Sabbia*, (5) *No Posidonia - Sedimenti vari*, and (6) *Mare Profondo*. Each colored line corresponds to the mean spectral response of one class, with the error bars representing the range of reflectance values observed within the selected samples.

The x-axis represents the 13 input layers used in the composite raster: Sentinel-2 bands B01, B11, B12, B02, B03, B04, B05, B06, B07, B08, B8A, B09, and the bathymetry layer. These were arranged in the specific order adopted during preprocessing. The y-axis corresponds to the reflectance values recorded for each band, expressed as scaled digital numbers. ArcGIS Pro calculates these values by averaging the pixel intensities from the selected training areas across all bands, which explains why the majority of values cluster in the range of approximately 1000–1200.

Overall, the profiles highlight key differences among the six classes. For instance, *Posidonia*-related classes show distinctive patterns in the visible and near-infrared regions compared with sandy or deep-water classes, which generally exhibit flatter responses. The inclusion of the bathymetric band at the end further enhances the ability to distinguish shallow benthic features from deeper marine areas. These spectral differences provided the physical foundation for the subsequent deep learning classification.

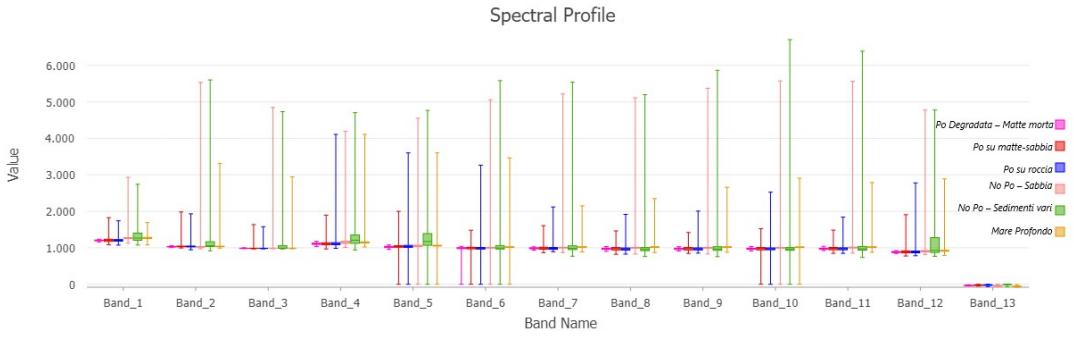


Figure 5.1: Spectral profiles extracted for six benthic classes in Capo Testa Island.

5.3 Performance metrics: definitions and computation

To evaluate the performance of the classification models, a set of standard metrics was considered. These metrics quantify how well the model predicted the class membership of each pixel, both during the training process and in the final evaluation stage. Since many readers may not be familiar with these measures, a short explanation is provided here.

5.3.1 Accuracy

Accuracy is one of the simplest and most widely used measures of classification performance. It is defined as the proportion of correctly classified pixels out of the total number of pixels:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

where TP and TN denote true positives and true negatives, and FP and FN denote false positives and false negatives, respectively.

Two types of accuracy are reported in this thesis:

- **Overall Accuracy**: the percentage of all pixels correctly classified across all classes.
- **Per-Class Accuracy**: the accuracy computed separately for each class, which highlights whether certain classes were more difficult for the model to predict.

5.3.2 Dice coefficient / F1 score

The Dice coefficient (also referred to as the F1 Score in classification tasks) is particularly useful in segmentation problems where class imbalance is common. It combines precision (the fraction of predicted positives that are correct) and recall (the fraction of true positives that were detected).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.3}$$

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

The F1 Score ranges from 0 (worst performance) to 1 (perfect classification). In this work, the Dice coefficient is equivalent to the F1 Score and was used to quantify segmentation quality.

5.3.3 Training and validation loss

During training, the model optimizes a loss function, which measures the difference between the predicted class probabilities and the ground truth labels. For semantic segmentation, the most common loss functions are variations of the cross-entropy loss.

The cross-entropy loss for one pixel can be written as:

$$\mathcal{L} = -\sum_{c=1}^{C} y_c \cdot \log(\hat{y}_c) \tag{5.5}$$

where C is the number of classes, y_c is the ground-truth label (1 if the pixel belongs to class c, 0 otherwise), and $\hat{y}_c$ is the predicted probability for class c.

Two curves are tracked during training:

- **Training Loss**: computed on the training dataset after each epoch, reflecting how well the model fits the data it has seen.
- **Validation Loss**: computed on a separate validation dataset, indicating how well the model generalizes to unseen data.

5.3.4 Epoch-level and final metrics

For each epoch of training, the model produces four key outputs: training loss, validation loss, accuracy, and F1 Score. These metrics evolve across epochs, allowing the detection of convergence, underfitting, or overfitting.

At the end of training, the model also reports **final performance metrics** (Precision, Recall, and F1 Score) for each class. These values are computed on the validation set by aggregating predictions across all images. In practice, this means that after the last epoch the model evaluates all validation pixels, counts the total TP, FP, and FN for each class, and then applies the formulas above. Thus, while the epoch-level curves show the evolution of performance during training, the final reported numbers represent a single, global evaluation across the full validation dataset.

5.4 Pixel classification using deep learning

In the previous chapter, the training of the deep learning model was described in detail. The outcome of that process for each reference year was a set of `.dlpk` files. A deep learning package (`.dlpk`) is an archive that contains three essential components:

- the trained network weights obtained after optimization,
- the model definition (architecture and metadata), and
- auxiliary information such as preprocessing parameters and class definitions.

In other words, the `.dlpk` encapsulates everything required for deployment of a trained model in ArcGIS Pro or other compatible environments. These packages are produced automatically when exporting a trained model from Python, ensuring reproducibility and portability.

In the previous chapter, the training of the deep learning model was described in detail. The outcome of that process for each reference year was a set of `.dlpk` files. A deep learning package (`.dlpk`) is an archive that contains three essential components:

- the trained network weights obtained after optimization,
- the model definition (architecture and metadata), and
- auxiliary information such as preprocessing parameters and class definitions.

In other words, the `.dlpk` encapsulates everything required for deployment of a trained model in ArcGIS Pro or other compatible environments. These packages are produced automatically when exporting a trained model from Python, ensuring reproducibility and portability.

As discussed in Section 4.4.1, semantic segmentation performs a pixel-wise classification of the image, producing a dense map in which every pixel is assigned to a class. An alternative approach would be **object-based image analysis (OBIA)**, which groups pixels into homogeneous objects and then assigns a class label to each object. Object-based methods are widely used in terrestrial remote sensing, especially when the goal is to classify land cover types with distinct boundaries such as agricultural fields, urban blocks, or forest stands.

However, this approach is less suitable for underwater environments. The boundaries of *Posidonia* meadows are often irregular and gradual rather than forming clear objects. In addition, the spectral differences between seagrass, sand, and algae can be very subtle, and the water column further reduces contrast. Segmenting the seafloor into meaningful objects under these conditions is challenging and can easily introduce classification errors. Pixel-based classification is therefore more appropriate, as it allows the model to exploit fine-scale spectral and textural information at the level of individual pixels. This enables a more precise delineation of meadow edges and small patches, which is essential for monitoring subtle changes in seagrass distribution over time.

5.4.1 Classify pixels using deep learning tool in ArcGIS Pro

Once the `.dlpk` was prepared, the next step was to apply it to the satellite images in ArcGIS Pro using the *Classify Pixels Using Deep Learning* tool (Esri, 2024). This geoprocessing tool takes a raster dataset as input, divides it into manageable tiles, applies the deep learning model to each tile, and assembles the results into a classified raster. Each output pixel is assigned to one of the predefined classes according to the network's prediction.

The tool provides several parameters that directly influence the quality and consistency of the classification:

- **Model (`.dlpk` file)**: the trained deep learning package produced during training. Each year had its own `.dlpk`, ensuring consistency between training and inference.
- **Tile Size**: defines the size of the image patches that the raster is divided into. The tile size must be consistent with what was used during training, otherwise the receptive field of the network may be incorrectly applied.
- **Batch Size**: determines how many tiles are processed at once during classification. This value should also remain compatible with the setting used during training to maintain computational efficiency and avoid memory errors.
- **Padding**: ensures that border pixels within each tile are classified correctly by including additional neighborhood context.

5.4.2 The role of padding

Padding is particularly important in convolutional neural networks because convolutions require neighboring pixels to compute a valid output. Without padding, edge pixels lack sufficient context, leading to reduced output size and possible misclassification along tile borders.

The output size O of a convolutional layer can be computed as (GeeksforGeeks, 2025):

$$O = \frac{(W - K + 2P)}{S} + 1 \tag{5.6}$$

where W is the input size, K is the kernel size, P is the padding, and S is the stride. If no padding is applied ($P = 0$), the output shrinks, while with padding the output can maintain the same size as the input.

Two main strategies are commonly used:

- **Valid Padding (no padding)**: convolution is only applied where the kernel fully overlaps the image. This reduces output size and discards edge pixels.

- **Same Padding (zero padding)**: additional pixels (typically zeros) are added around the image border so that the output size matches the input size, preserving edge information.

Figure 5.2 illustrates this principle. The top row shows a convolution on a 6×6 input with a 3×3 kernel and stride $= 1$, without padding. The result is a smaller 4×4 output where edge pixels are ignored. The bottom row shows the same input with padding $= 1$. Here, the image expands to 8×8 before convolution, and the output maintains the original 6×6 size while preserving border information.

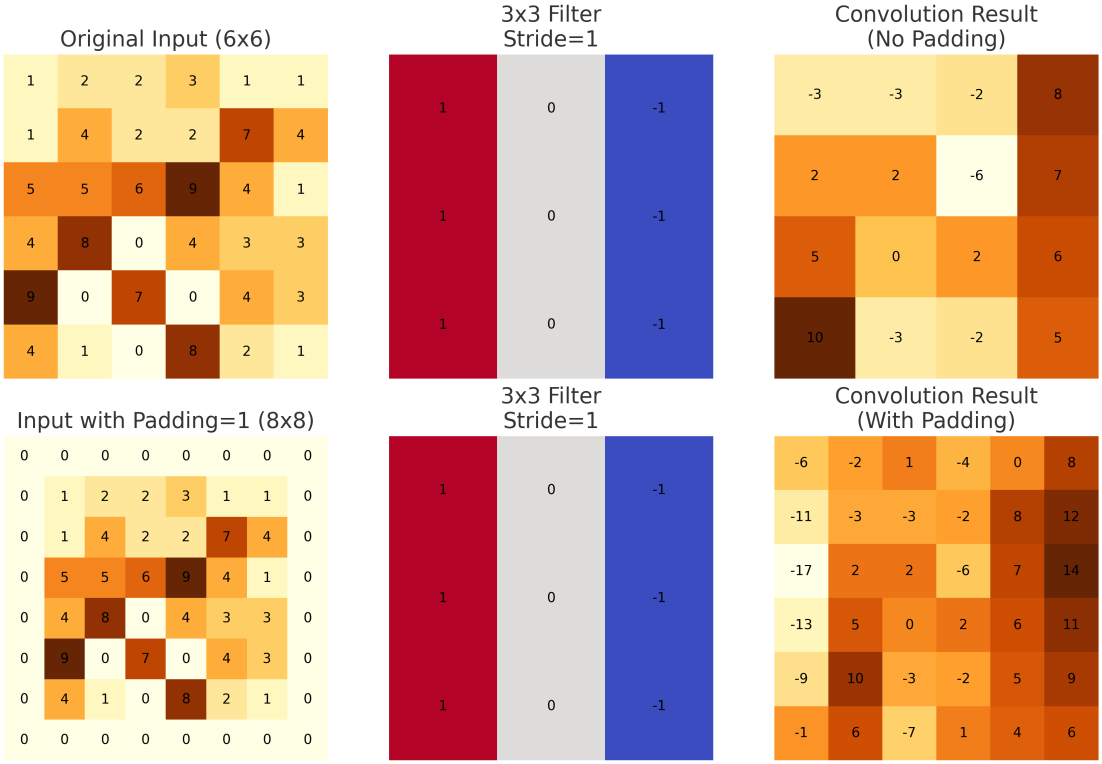


Figure 5.2: Effect of padding on convolution. Adapted from GeeksforGeeks (GeeksforGeeks, 2025).

In the ArcGIS Pro implementation, padding is applied at the tile level. Large rasters are divided into smaller tiles for processing, but without padding, predictions at tile edges would be based on incomplete context, leading to visible seams in the final mosaic. By adding padding, each tile is expanded so that all border pixels are predicted with full neighborhood information. After inference, the padded border is discarded, and the central predictions are retained, producing a seamless classification across the entire raster.

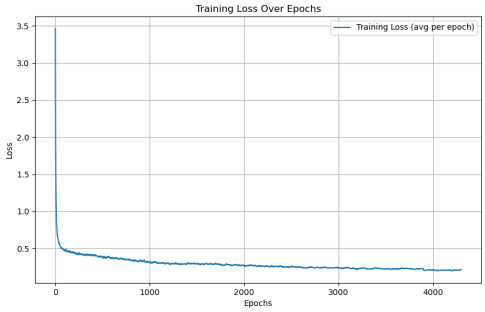
5.5 Model training and classification results

5.5.1 Results for 2015

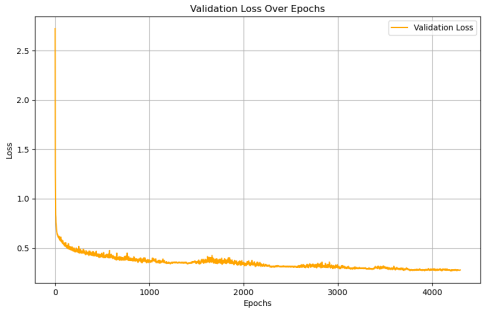
5.5.1.1 Training and Validation Curves

The evolution of the training process for 2015 is illustrated in Figure 5.3, which reports the training loss, validation loss, accuracy, and Dice Score (F1). These curves reflect both the convergence properties of the network and its capacity to generalize.

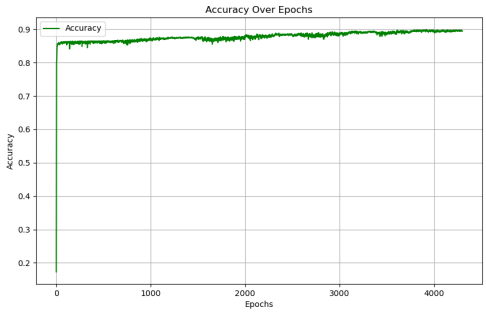
The training loss decreased sharply during the first few hundred epochs, stabilizing at values around 0.25. This pattern indicates efficient early learning, consistent with the ability of convolutional layers to rapidly capture low-level spatial features. The validation loss followed a nearly identical trend and reached a similar plateau, which suggests that the model did not overfit substantially to the training data. The absence of divergence between the two loss curves confirms stable optimization and adequate regularization.



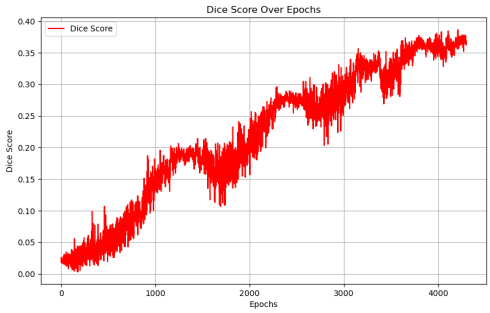
(a) Training loss



(b) Validation loss



(c) Accuracy



(d) Dice Score (F1)

Figure 5.3: Training and validation performance curves for the 2015 dataset.

Accuracy improved rapidly to above 85% and then increased gradually to approach 90%. However, this metric primarily reflects the dominance of background or majority classes, and therefore must be interpreted with caution in an imbalanced classification problem. For this reason, the Dice Score provides a more meaningful indicator of

segmentation quality.

The Dice Score evolved more gradually than accuracy and reached values close to 0.38 after 4000 epochs. While this value is lower than the overall accuracy, it provides a more realistic measure of segmentation quality because it balances precision and recall, which is particularly important for minority classes such as fragmented *Posidonia* patches.

Interestingly, the Dice curve did not rise monotonically but showed phases of decline followed by renewed increases. This behavior corresponds to the learning rate schedule applied during training (Table 4.3). Each time the learning rate was smoothly reduced (from 4.0×10^{-4} down to 5.0×10^{-5}), the optimizer effectively "reset" its step size. At the beginning of each new phase, the Dice Score often dropped slightly as the model adjusted to the smaller learning rate, before improving again as the optimization stabilized.

This effect is typical when staged learning rates are used: higher values help the model explore the loss landscape broadly, while lower values encourage finer adjustments. The temporary decreases in Dice reflect the transition between these two regimes rather than a loss of generalization. Over the long run, the gradual upward trend indicates that the model benefited from the schedule, improving its ability to delineate class boundaries despite the inherent noise of underwater imagery. The fluctuations observed are expected in ecological classification tasks, where class imbalance and spectral overlap make pixel-level segmentation highly variable.

Taken together, the curves demonstrate that the network achieved stable convergence without evidence of severe overfitting. The moderate Dice values highlight the difficulty of distinguishing spectrally similar benthic substrates, but the consistent improvement across epochs suggests that the trained model captured relevant features for subsequent classification of the 2015 imagery.

5.5.1.2 Classified raster map

The classified raster for 2015, together with the corresponding ground truth map, is presented in Figure 5.4. The model was trained using the configuration summarized in Table 5.1, and the final output achieved an overall accuracy of 0.8825 with a Kappa coefficient of 0.8433.

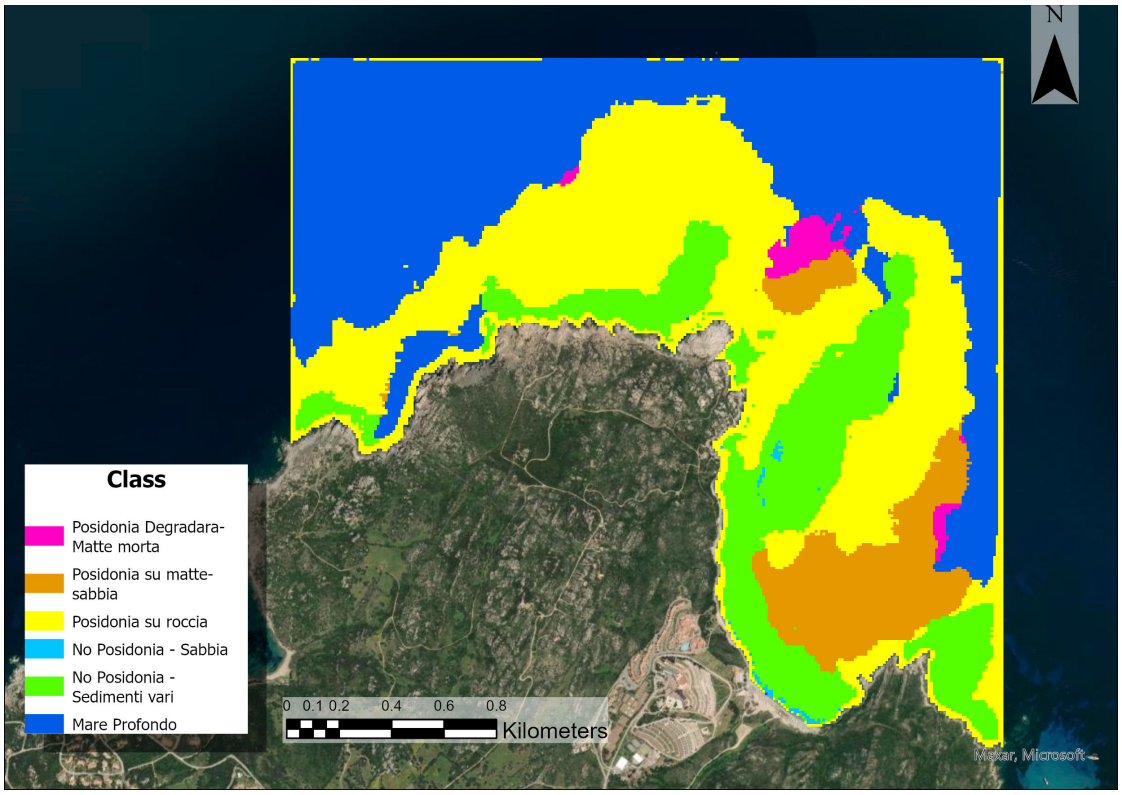
Table 5.1: Training settings and performance metrics for the 2015 classification.

Parameter	Value
Architecture	DeepLabv3
Backbone	ResNet-34
Tile size	30
Stride	15
Padding	4
Validation split	0.20
Oversampled classes	[1, 2, 3]
Class weights	(40.8152, 4.9418, 0.4318, 30.0506, 0.7893, 6.006)
Accuracy	0.8825
Kappa	0.8433

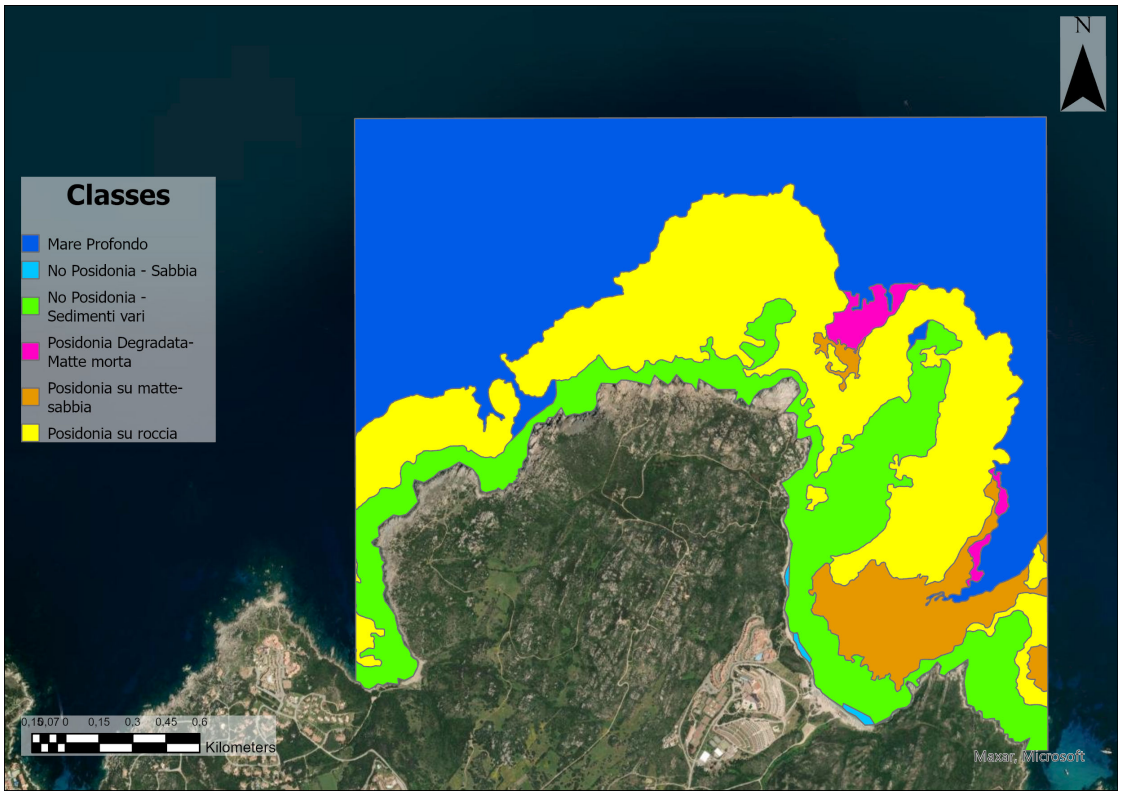
Each pixel in the study area was assigned to one of the six predefined classes. The classification reveals the spatial structure of *P.O.* meadows along the Capo Testa coastline, as well as the distribution of sandy and sedimentary substrates. The class *Mare Profondo* (dark blue) dominates the deeper offshore zones, where water depth prevents seagrass development and reduces the effectiveness of optical detection. Moving toward the coastline, extensive areas were classified as *Posidonia su roccia* (yellow), indicating meadows attached to rocky substrates, which is consistent with the geomorphology of Capo Testa.

Broad patches of *Posidonia su matte–sabbia* (orange) were also identified, marking areas where meadows colonize sandy or matte substrates. Smaller but distinct regions of*Posidonia Degradata – Matte morta* (magenta) are visible, corresponding to degraded or partially regressed meadow structures. The non-*Posidonia* classes are mainly concentrated in transitional zones: *No Posidonia – Sabbia* (light blue) outlines shallow sandy areas along the coast, while *No Posidonia – Sedimenti vari* (green) appears more fragmented and interspersed with vegetated zones, reflecting the natural heterogeneity of the seabed.

Overall, the 2015 classification depicts a heterogeneous coastal landscape composed of dense *Posidonia* meadows, degraded patches, and unvegetated substrates. The comparison with the ground truth map highlights both the strengths and the limitations of the model in separating spectrally similar benthic classes, particularly in shallow transitional areas.



(a) Classified raster (2015).



(b) Ground truth model.

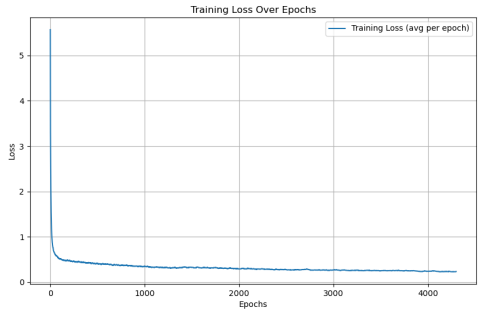
Figure 5.4: Classified raster for 2015 compared with the ground truth map.

5.5.2 Results for 2018

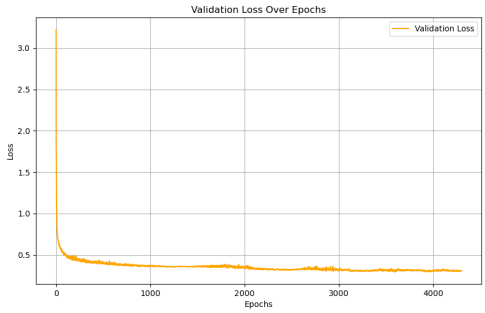
5.5.2.1 Training and validation curves

The training process for 2018 (Figure 5.5) showed stable convergence, with both training and validation losses decreasing smoothly and reaching plateaus without large divergence. Accuracy remained consistently high, close to 88–89%, very similar to the 2015 run.

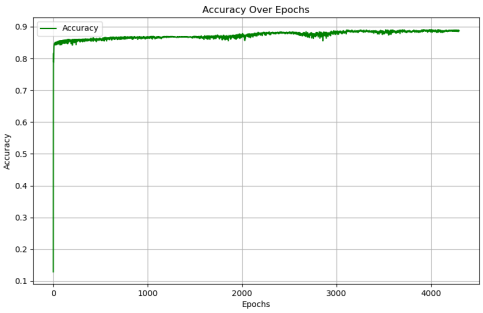
The Dice Score, however, plateaued at slightly lower values than in 2015, around 0.34–0.35. This suggests that while the classifier maintained strong overall accuracy, it struggled more with the finer balance between precision and recall in minority classes. The curve also showed more pronounced fluctuations after epoch 2000, which may be linked to spectral variability in the 2018 imagery (e.g., water clarity or acquisition conditions).



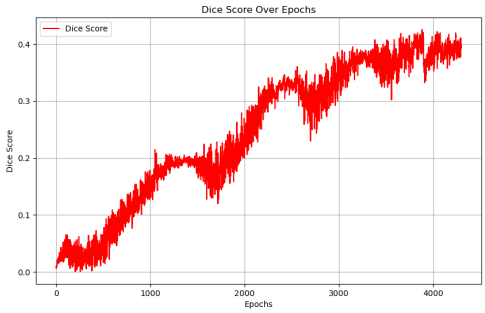
(a) Training loss



(b) Validation loss



(c) Accuracy



(d) Dice Score (F1)

Figure 5.5: Training and validation performance curves for the 2018 model.

5.5.2.2 Classified raster map

The classified raster for 2018 is shown in Figure 5.6. The best classification for this year was obtained with the settings in Table 5.2, yielding an overall accuracy of 0.8667 and a kappa of 0.8217.

Table 5.2: Training settings and performance metrics for the 2018 classification.

Parameter	Value
Architecture	DeepLabv3
Backbone	ResNet-34
Batch size	16
Tile (chip) size	30
Stride	15
Padding	14
Validation split	0.20
Oversampled classes	[1, 2, 3]
Class weights	(40.8152, 5.9418, 1.4318, 28.0506, 0.8893, 8.006)
Accuracy	0.8667
Kappa	0.8217

A visible artifact in this product is the narrow stripe along the northern frame. This is a perimeter tiling effect, not a seabed feature. When a tile touches the scene boundary, the network does not receive full spatial context outside the image. That missing context, together with zero padding at the edge, can bias predictions and produce a uniform band that is wrongly labeled. During the first iterations of the study, such seamlines appeared on several tile borders and degraded the map quality. We reduced these artifacts by using overlap (stride smaller than chip), setting a larger padding (14), and prioritizing the central area of each tile when mosaicking the outputs. The residual stripe in 2018 is much smaller than in early trials, and in later years it is nearly absent.

Separately, errors were also present at true class boundaries, driven by mixed pixels at 10 m, limited context near transitions, and class imbalance. Tuning the class weights and using padding helped sharpen these boundaries, which explains the cleaner edges seen in later results.

From an ecological perspective, the 2018 map is broadly consistent with 2015 but with some local changes. The extent of *Posidonia su matte-sabbia* (orange) expanded in the southern sector, while areas of *Posidonia Degradata - Matte morta* (magenta) appear more continuous in the northeast. Shallow rocky meadows (*Posidonia su roccia*, yellow) remain dominant along the coastline, although boundaries with *No Posidonia - Sabbia* (light blue) are less sharp than in 2015, consistent with the slightly lower accuracy. Fragmented patches of *No Posidonia - Sedimenti vari* (green) persist between sandy and vegetated zones, reflecting small-scale seabed variability.

Overall, the 2018 classification confirms that the model can separate the main classes while also showing the two main sources of error that guided our later

improvements: perimeter tiling artifacts at the scene edge and confusion at true class boundaries. Targeted adjustments to overlap, padding, and class weights were essential to reduce both effects in subsequent years.

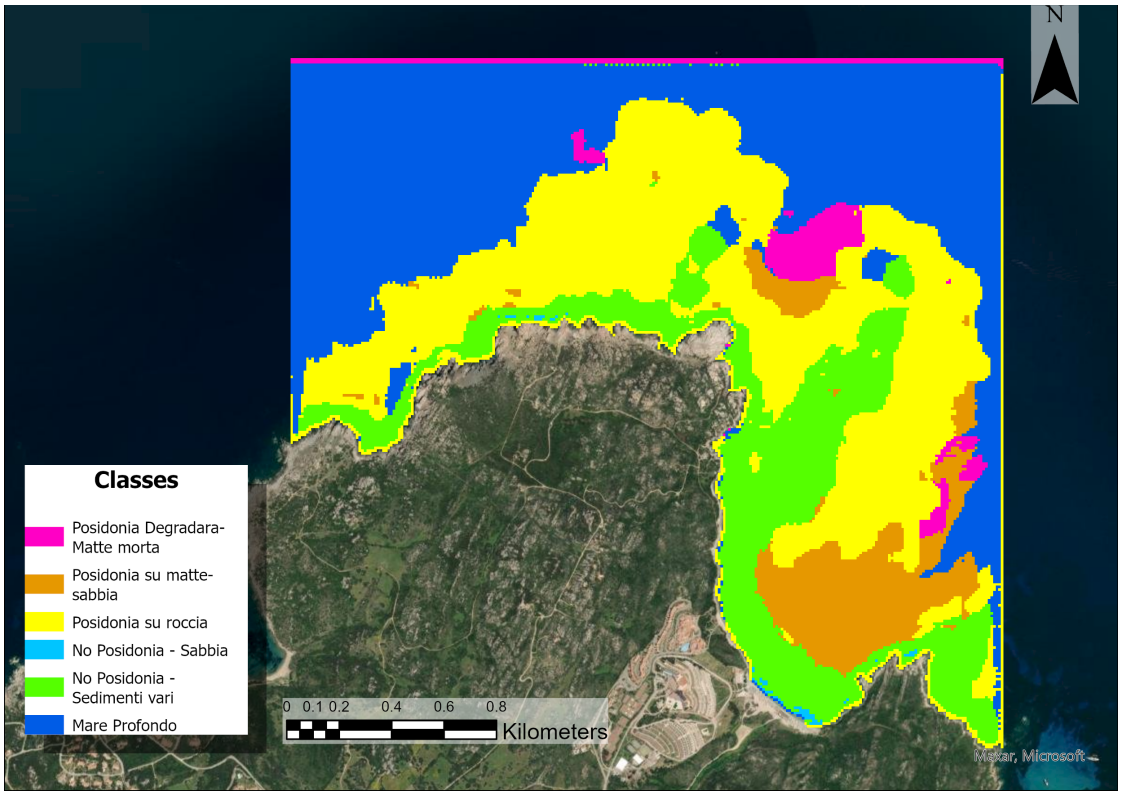


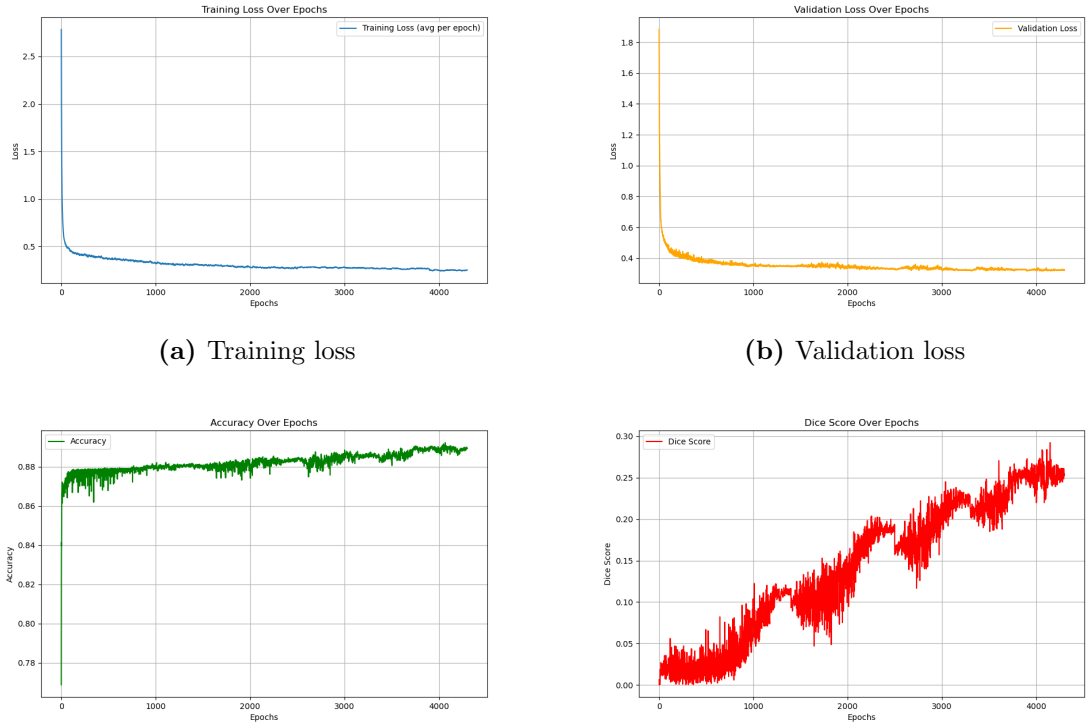
Figure 5.6: Classified raster for 2018 showing the spatial distribution of six benthic classes in Capo Testa Island.

5.5.3 Results for 2021

5.5.3.1 Training and validation curves

The training dynamics in 2021 (Figure 5.7) show a slightly different behavior compared with the earlier years. Both training and validation losses reached stable plateaus again, but the Dice Score ended lower, around 0.26. This is noticeably weaker than 2015 and 2018, and it points to difficulties in segmenting minority classes. In fact, the Dice curve kept climbing slowly across the epochs but never reached the levels of previous runs.

Accuracy, on the other hand, stayed consistently high (around 88–89%), which again confirms the imbalance issue: the network predicted the dominant classes reliably but struggled with finer separations between seagrass and non-vegetated sediments. This year's curves make it clear that overall accuracy alone would give a misleading impression of performance.



(a) Training loss

(b) Validation loss

(c) Accuracy

(d) Dice Score (F1)

Figure 5.7: Training and validation performance curves for the 2021 model.

5.5.3.2 Classified raster map

The 2021 classification (Figure 5.8) was produced with the settings reported in Table 5.3, which led to an overall accuracy of 0.8697 and a Kappa coefficient of 0.8261.

Table 5.3: Training settings and performance metrics for the 2021 classification.

Parameter	Value
Architecture	DeepLabv3
Backbone	ResNet-34
Batch size	16
Tile (chip) size	30
Stride	15
Padding	14
Validation split	0.20
Oversampled classes	[1, 2, 3]
Class weights	(10.8023, 2.6791, 0.8081, 30.7181, 0.8919, 0.6144)
Accuracy	0.8698
Kappa	0.8261

The map depicts a different balance of classes compared with 2015 and 2018. Areas

labeled as *Posidonia su matte-sabbia* (orange) expanded considerably in the southeast, while *No Posidonia - Sedimenti vari* (green) became more widespread across the central part of the bay. Interestingly, the patches of *Posidonia Degradata - Matte morta* (magenta) shrank compared with previous years, suggesting either genuine regression of degraded zones or classification bias.

Rocky meadows (*Posidonia su roccia*, yellow) still dominate shallow zones, but their boundaries with sandy classes (light blue) appear less sharply defined, which may be a reflection of the weaker Dice performance noted in the training results.

Overall, the 2021 raster shows a shift in class proportions, with more heterogeneous non-Posidonia substrates and less degraded matte. This will be important to cross-check against field references and the accuracy assessment in the next section.

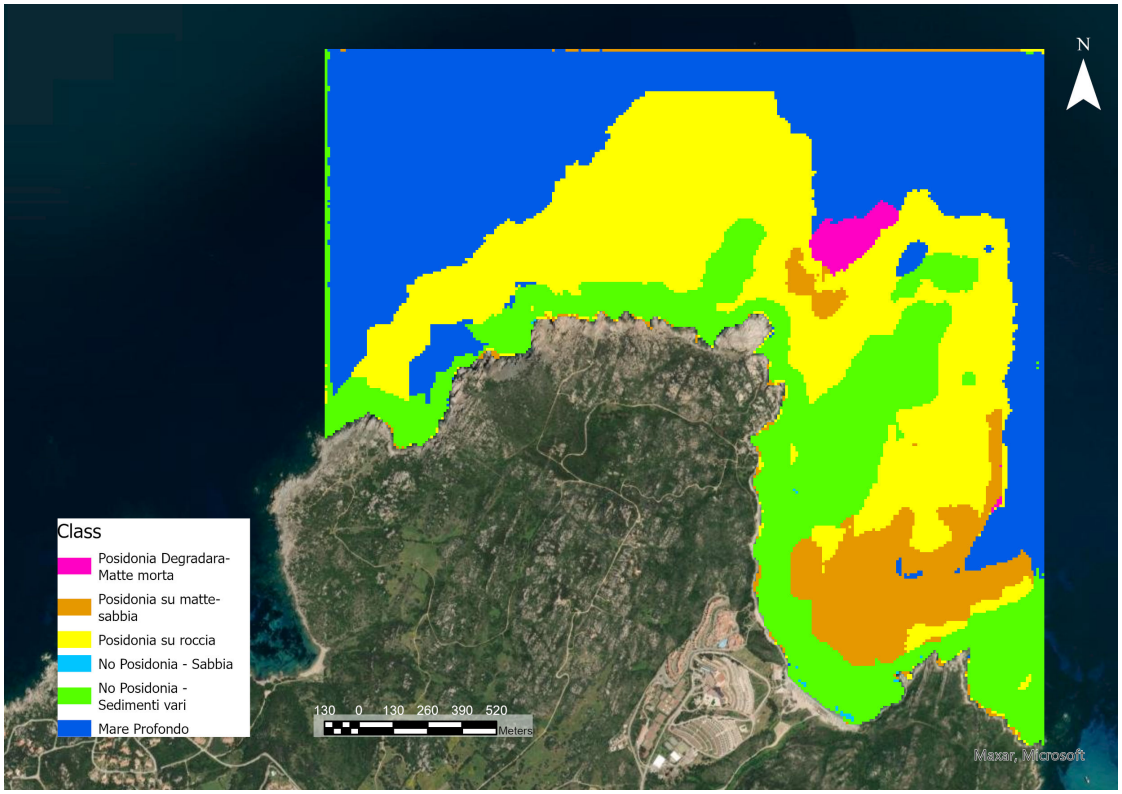


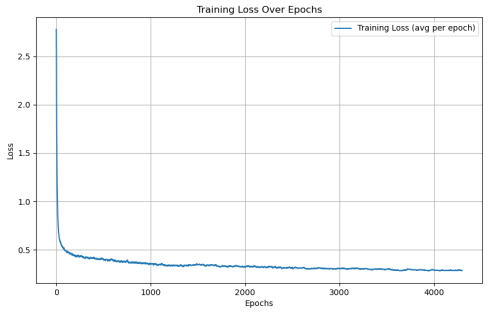
Figure 5.8: Classified raster for 2021 showing the spatial distribution of six benthic classes in Capo Testa Island.

5.5.4 Results for 2024

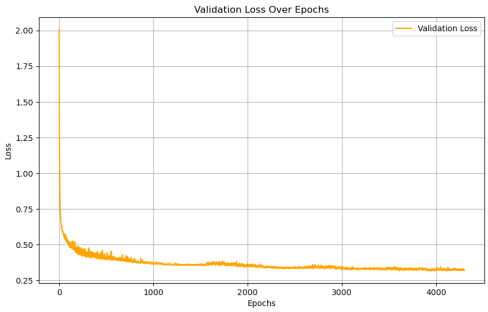
5.5.4.1 Training and validation curves

The 2024 curves (Figure 5.9) show that the model converged smoothly, with both training and validation losses decreasing toward stable values and no evidence of divergence. Accuracy remained close to 88% throughout, while the Dice Score plateaued at about 0.28, slightly higher than 2021 but still below the values obtained in 2015.

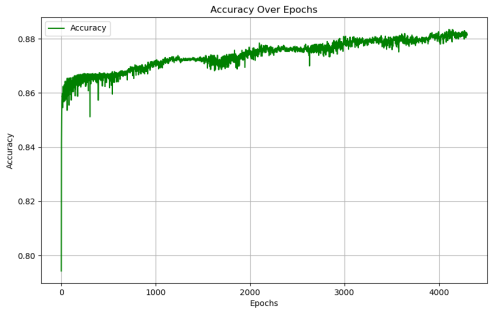
One noticeable improvement in this run was the stability of the loss curves: fluctuations were smaller than in 2018 and 2021, indicating that the optimization process was more consistent. The Dice Score continued to increase steadily across epochs, suggesting the model was still learning finer boundary representations late in training.



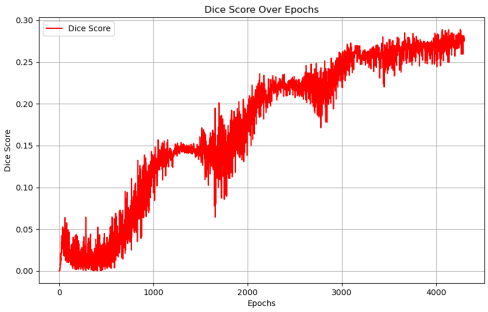
(a) Training loss



(b) Validation loss



(c) Accuracy



(d) Dice Score (F1)

Figure 5.9: Training and validation performance curves for the 2024 model.

5.5.4.2 Classified raster map

The 2024 classification (Figure 5.10) was generated using the settings listed in Table 5.4, which provided the best overall performance of the study. The model achieved an accuracy of 0.8889 and a Kappa coefficient of 0.8504, both higher than in previous years.

In addition to the numerical results, a clear qualitative improvement is evident in the map itself. Unlike the 2015 and 2018 outputs, the 2024 raster no longer exhibits the misclassified "border seams" at the edges of the tiles. The padding strategy proved effective, eliminating these artifacts and yielding smoother, more natural class transitions.

Spatially, the classification shows patterns broadly consistent with earlier years but with sharper boundaries. Large areas of *Posidonia su roccia* (yellow) continue to dominate shallow coastal zones, while *Posidonia su matte–sabbia* (orange) remains concentrated in the southeast. Fragmented patches of *No Posidonia – Sedimenti vari*

(green) are still scattered across the bay, and degraded matte (magenta) is present but more localized.

Overall, the 2024 raster appears cleaner and more reliable than previous outputs, making it a particularly robust basis for multi-temporal comparison of seagrass dynamics.

Table 5.4: Training settings and performance metrics for the 2024 classification.

Parameter	Value
Architecture	DeepLabv3
Backbone	ResNet-34
Batch size	16
Tile (chip) size	30
Stride	15
Padding	14
Validation split	0.20
Oversampled classes	[1, 2, 3]
Class weights	(15.2898, 2.8490, 0.9003, 33.6082, 0.9633, 0.9616)
Accuracy	0.8889
Kappa	0.8504

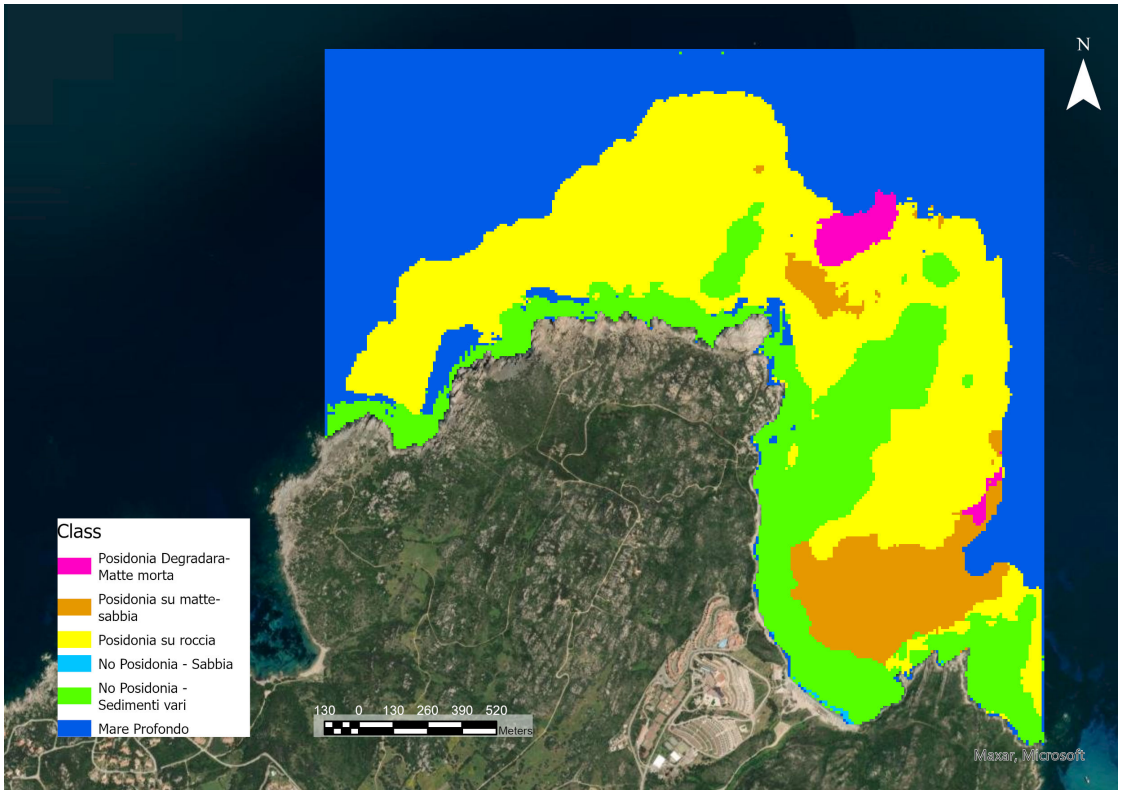


Figure 5.10: Classified raster for 2024 showing the spatial distribution of six benthic classes in Capo Testa Island.

5.6 Accuracy assessment of classification

To evaluate how reliable the classification outputs were, we performed an accuracy assessment, which is one of the most common and recommended methods for thematic map validation. Accuracy assessment measures the quality of the classified map by comparing it against a reference dataset, in our case the ground truth model. The comparison is summarized in a Classification Error Matrix, also known as a Confusion Matrix (Maxwell et al., 2019).

5.6.1 Workflow in ArcGIS Pro

Before starting the accuracy assessment, a preprocessing step was necessary. The classified rasters generated by the deep learning model were stored as 32-bit images, which are not compatible with the accuracy assessment tools in ArcGIS Pro. To resolve this, the *Reclassify* tool was applied to convert the pixel depth from 32-bit to 8-bit integers. This step ensured that each class was represented by a unique integer value, making the raster suitable for subsequent accuracy evaluation.

Once the raster was reclassified, the workflow proceeded through three main stages using tools from the ArcGIS Pro Image Analyst toolbox:

1. **Create Accuracy Assessment Points**: this tool was applied to the reclassified raster to generate a random stratified sample of points (ESRI, 2025b). We set the parameter to generate 300 points, distributed across the classes, although the final output contained 315 points due to the internal balancing algorithm of ArcGIS Pro. The output was a point layer (Figure 5.11) with an attribute table that contained two key fields:

    - `Classified`: the class value of the pixel in which the point was located, assigned directly from the reclassified raster. This field contained integer codes corresponding to the thematic classes.
    - `GrndTruth`: initially set to -1 for all points, representing an undefined reference class.

2. **Update Accuracy Assessment Points**: the `GrndTruth` field was updated using the ground truth shapefile. To automate this step, a Python attribute rule was implemented in ArcGIS Pro (Data Design → Attribute Rules). This script ensured that each point inherited the correct reference class label from the ground truth dataset. At the end of this step, each point had two values: one representing the predicted class (`Classified`) and one representing the actual class (`GrndTruth`).

3. **Compute Confusion Matrix**: finally, the two fields (`Classified` vs. `GrndTruth`) were compared using the *Compute Confusion Matrix* tool. This produced a classification error matrix that summarized the correctly classified points (diagonal values) and misclassifications (off-diagonal values), from which accuracy indices were derived.

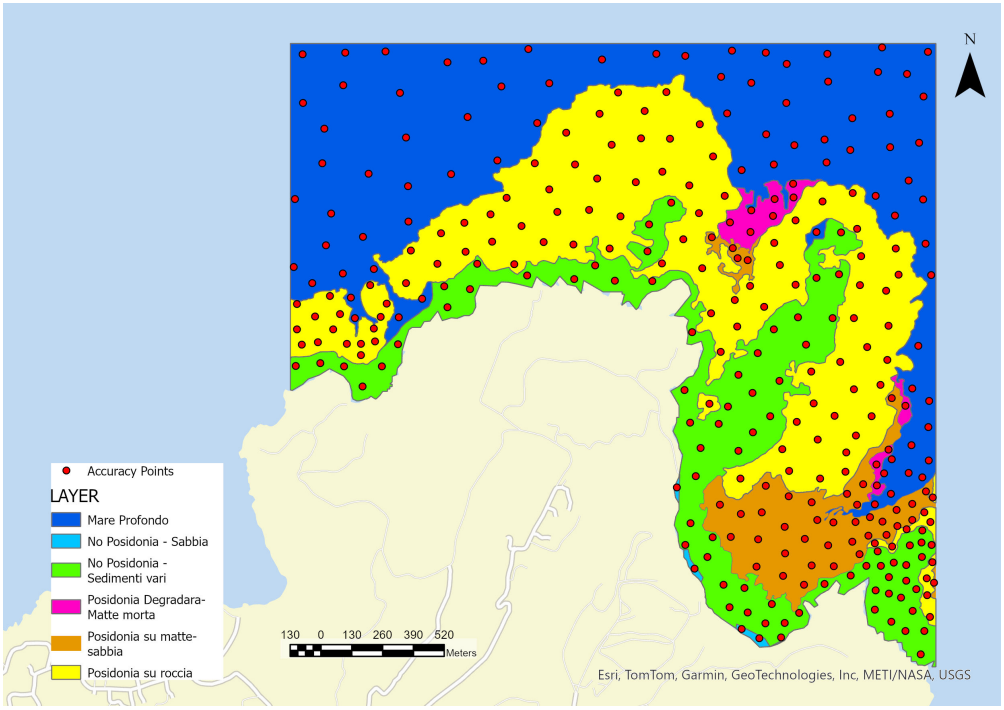


Figure 5.11: Accuracy assessment points generated with the *Create Accuracy Assessment Points* tool in ArcGIS Pro.

5.6.2 The confusion matrix concept

The confusion matrix is a square table in which rows represent the ground truth (reference) data and columns represent the predicted classes. Correct classifications are found along the diagonal of the matrix, while off-diagonal values represent misclassifications.

From this matrix, several indices can be derived:

- **User's Accuracy (U_Accuracy)**: the proportion of correctly classified samples among all samples assigned to a given class (sensitive to commission error).
- **Producer's Accuracy (P_Accuracy)**: the proportion of correctly classified samples among all reference samples of a given class (sensitive to omission error).
- **Overall Accuracy (OA)**: the total proportion of correctly classified samples

across all classes. It is defined as:

$$OA = \frac{\sum_{i=1}^{k} n_{ii}}{N} \tag{5.7}$$

where n_{ii} are the diagonal elements (correct classifications), k is the number of classes, and N is the total number of samples.

- **Kappa Coefficient (κ)**: a measure of agreement between classification and reference data adjusted for random chance. It is defined as:

$$\kappa = \frac{N \sum_{i=1}^{k} n_{ii} - \sum_{i=1}^{k} (n_{i+} \cdot n_{+i})}{N^2 - \sum_{i=1}^{k} (n_{i+} \cdot n_{+i})} \tag{5.8}$$

where n_{i+} is the total number of samples in row i (reference class i) and n_{+i} is the total number of samples in column i (predicted class i).

Results for 2015

The accuracy assessment for the 2015 classification is presented in Table 5.5. The confusion matrix compares the predicted classes (columns) with the ground truth reference classes (rows). Values on the diagonal represent correctly classified samples, while off-diagonal values correspond to misclassifications.

Table 5.5: Confusion matrix and accuracy indices for 2015.

ClassValue	C_1	C_2	C_3	C_4	C_5	C_6	Total	U_Accuracy	Kappa
C_1	10	0	0	0	0	1	11	0.9090	0
C_2	0	36	6	0	0	0	42	0.8571	0
C_3	0	3	91	1	8	6	109	0.8348	0
C_4	0	0	0	3	1	0	4	0.7500	0
C_5	0	0	2	0	65	0	67	0.9701	0
C_6	1	1	5	0	2	73	82	0.8902	0
Total	11	40	104	4	76	80	315	0	0
P_Accuracy	0.9090	0.9000	0.8750	0.7500	0.8552	0.9125	0	0.8825	0
Kappa	0	0	0	0	0	0	0	0	0.8433

The overall accuracy for 2015 was **88.3%**, with a Kappa coefficient of **0.84**, which indicates strong agreement between the classification and the ground truth dataset.

To illustrate how to read the confusion matrix, consider the class *C3: Posidonia su roccia*. The reference dataset contained 104 points for this class (row total). Of these, 91 points were correctly classified as C3 (diagonal entry), while the remaining points were misclassified: 3 points as C2 (*Posidonia su matte–sabbia*), 1 point as C4 (*No Posidonia – Sabbia*), 8 points as C5 (*No Posidonia – Sedimenti vari*), and 6 points as C6 (*Mare Profondo*). In other words, the row shows how the reference class was distributed across predicted categories.

Columns, on the other hand, represent how many points the classifier assigned to each predicted class. For example, column C3 contains 104 points predicted as *Posidonia su roccia*, but only 91 of them were truly C3 in the ground truth. This explains why User's Accuracy for C3 is 83.5% (proportion of correctly classified points among all predictions for that class), while Producer's Accuracy for C3 is 87.5% (proportion of correctly identified points among all ground truth samples for that class).

Among all classes, *No Posidonia – Sedimenti vari* (C5) showed the highest User's Accuracy (97.0%), while *No Posidonia – Sabbia* (C4) performed worst, with both User's and Producer's Accuracies around 75.0%. This reflects the challenge of distinguishing sandy areas from spectrally similar mixed substrates.

Results for 2018

The accuracy assessment for the 2018 classification is presented in Table 5.6. The confusion matrix compares the classified outputs (columns) with the ground truth reference classes (rows). Correct classifications are on the diagonal, while off-diagonal entries represent misclassifications.

Table 5.6: Confusion matrix and accuracy indices for 2018.

ClassValue	C_1	C_2	C_3	C_4	C_5	C_6	Total	U_Accuracy	Kappa
C_1	9	0	1	0	0	2	12	0.7500	0
C_2	2	33	2	0	0	2	39	0.8462	0
C_3	0	5	91	0	7	4	107	0.8505	0
C_4	0	0	0	1	0	0	1	1.0000	0
C_5	0	0	0	2	67	0	69	0.9710	0
C_6	0	2	10	1	2	72	87	0.8276	0
Total	11	40	104	4	76	80	315	0	0
P_Accuracy	0.8182	0.8250	0.8750	0.2500	0.8816	0.9000	0	0.8667	0
Kappa	0	0	0	0	0	0	0	0	0.8217

The overall accuracy for 2018 was **86.7%**, with a Kappa coefficient of **0.82**, indicating strong agreement between the classification and the reference dataset.

If we ignore C4 which is not well represented, the best performance was observed for *C5: No Posidonia – Sedimenti vari*, with a User's Accuracy of 97.1% and a Producer's Accuracy of 88.2%. This suggests that this class was both reliably detected when present and rarely confused with others. *C3: Posidonia su roccia* also showed good performance, with Producer's Accuracy of 87.5% and User's Accuracy of 85.0%.

On the other hand, *C4: No Posidonia – Sabbia* had very limited representation in the reference data (only one point) and therefore achieved an artificially perfect User's Accuracy of 100%, but with little statistical significance. *C1: Posidonia*

Degradata – Matte morta showed lower reliability, with User's Accuracy of 75.0% due to misclassifications into *Mare Profondo* (C6).

Overall, the 2018 classification performed slightly weaker than 2015, particularly for minority classes, but still maintained robust agreement for the dominant categories.

Results for 2021

Table 5.7 shows the confusion matrix for 2021. As before, rows correspond to ground truth classes and columns to the classified outputs.

Table 5.7: Confusion matrix and accuracy indices for 2021.

ClassValue	C_1	C_2	C_3	C_4	C_5	C_6	Total	U_Accuracy	Kappa
C_1	7	0	0	0	0	0	7	1.0000	0
C_2	2	34	1	0	0	0	37	0.9189	0
C_3	0	4	82	0	0	5	91	0.9011	0
C_4	0	0	0	1	0	0	1	1.0000	0
C_5	0	1	11	3	76	1	92	0.8261	0
C_6	2	1	10	0	0	74	87	0.8506	0
Total	11	40	104	4	76	80	315	0	0
P_Accuracy	0.6364	0.8500	0.7885	0.2500	1.00	0.9250	0	0.8698	0
Kappa	0	0	0	0	0	0	0	0	0.8261

The 2021 classification reached an overall accuracy of **87.0%** with a Kappa coefficient of **0.83**, confirming solid agreement between predictions and ground truth.

Unlike 2015 and 2018, performance was more uneven across classes. Some categories, such as *C1: Posidonia Degradata – Matte morta* and *C4: No Posidonia – Sabbia*, were perfectly matched to the reference data, but this is partly due to their small sample sizes (7 and 1 points, respectively). Larger classes give a clearer picture: *C2: Posidonia su matte–sabbia* and *C3: Posidonia su roccia* both performed strongly, with User's Accuracies above 90%.

By contrast, *C5: No Posidonia – Sedimenti vari* showed more misclassifications, especially into vegetated classes, and achieved only 82.6% User's Accuracy. Similarly, *C6: Mare Profondo* was sometimes confused with rocky areas (C3), lowering its reliability to about 85%.

In summary, the 2021 results were consistent with previous years in terms of overall accuracy, but they highlighted the persistent difficulty of distinguishing mixed sedimentary zones from adjacent seagrass or rocky habitats. These patterns underscore the role of spectral overlap in reducing classification precision for transitional seabed types.

Results for 2024

Table 5.8 presents the confusion matrix for 2024. Rows are reference (ground truth) classes, and columns are the predicted outputs.

Table 5.8: Confusion matrix and accuracy indices for 2024.

ClassValue	C_1	C_2	C_3	C_4	C_5	C_6	Total	U_Accuracy	Kappa
C_1	7	0	0	0	0	0	7	1.0000	0
C_2	1	33	0	0	0	1	35	0.9429	0
C_3	1	4	94	0	4	4	107	0.8785	0
C_4	0	0	0	1	0	0	1	1.0000	0
C_5	0	0	2	3	70	0	75	0.9333	0
C_6	2	3	8	0	2	75	90	0.8333	0
Total	11	40	104	4	76	80	315	0	0
P_Accuracy	0.6364	0.8250	0.9038	0.2500	0.9211	0.9375	0	0.8889	0
Kappa	0	0	0	0	0	0	0	0	0.8504

The 2024 classification achieved an overall accuracy of **88.9%** with a Kappa coefficient of **0.85**, which is the strongest agreement across all four years.

Performance was particularly high for *C5: No Posidonia – Sedimenti vari* and *C2: Posidonia su matte–sabbia*, both with User's Accuracies above 93% and Producer's Accuracies exceeding 82%. This suggests that by 2024, the model was more reliable in separating mixed substrates from adjacent classes.

C3: Posidonia su roccia also performed well, with a Producer's Accuracy above 90%, though still subject to confusion with *Mare Profondo* (C6). Indeed, most of the remaining misclassifications occurred between C3 and C6, reflecting the spectral similarity of rocky seabeds at greater depths.

Minor classes such as *C1: Posidonia Degradata – Matte morta* and *C4: No Posidonia – Sabbia* again appeared perfectly classified, though the small number of reference points makes these values less representative statistically.

In summary, the 2024 results confirmed robust model performance, with more balanced accuracies across the main benthic classes and fewer boundary effects compared to previous years.

5.7 Multi-temporal change detection

5.7.1 Methodological approach

To investigate temporal shifts in benthic habitats, and particularly the distribution of P.O., a post-classification change detection was performed in ArcGIS Pro using the Change Detection Wizard with the categorical change option (ESRI, 2025a). This method directly compares thematic rasters from different years on a pixel-by-pixel

basis, quantifying which areas remained stable and which transitioned from one class to another.

The analysis was carried out for three consecutive periods (2015–2018, 2018–2021, 2021–2024) as well as for the full interval from 2015 to 2024. The choice of a post-classification approach ensured that differences between years were attributed to genuine class transitions rather than to sensor-specific spectral variability.

5.7.2 Change maps

Figures 5.12 to 5.14 illustrate the detected changes for each period, with each color corresponding to a specific transition between benthic classes. These maps allow a visual assessment of where seagrass meadows have contracted, expanded, or remained stable, and highlight specific coastal sectors where Posidonia has been replaced by sand or mixed sediments.

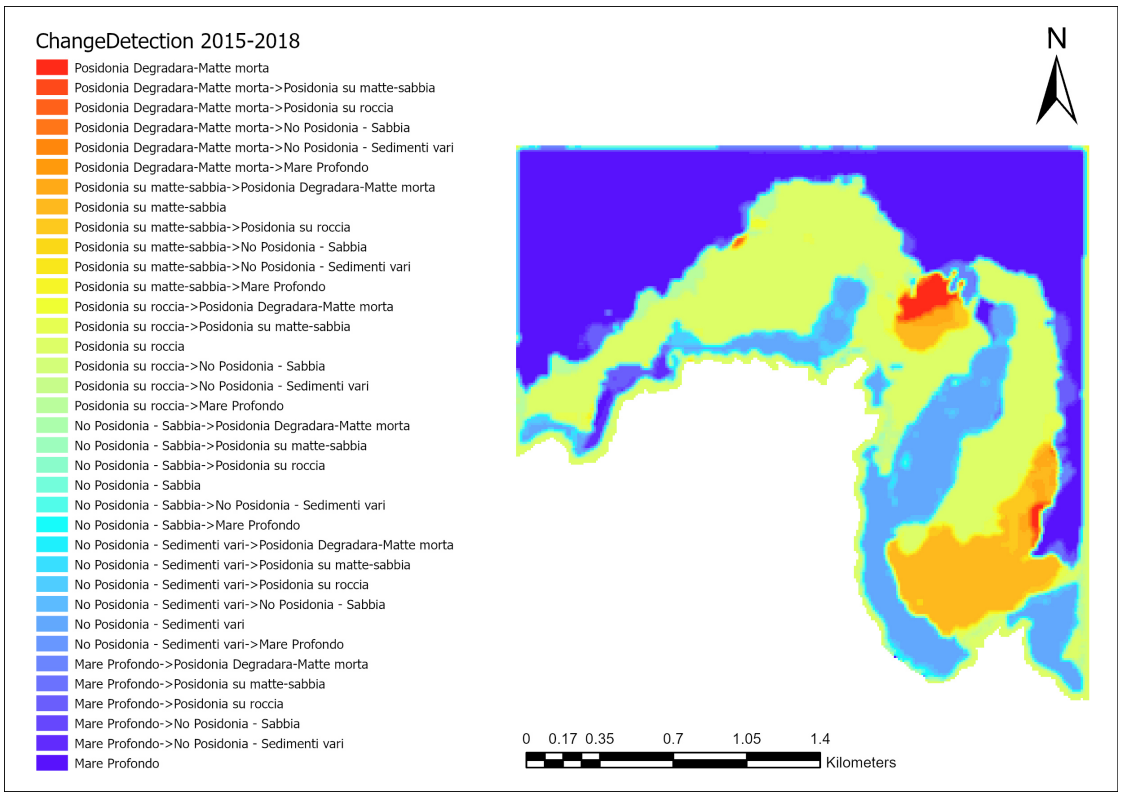


Figure 5.12: Change detection map for 2015–2018.

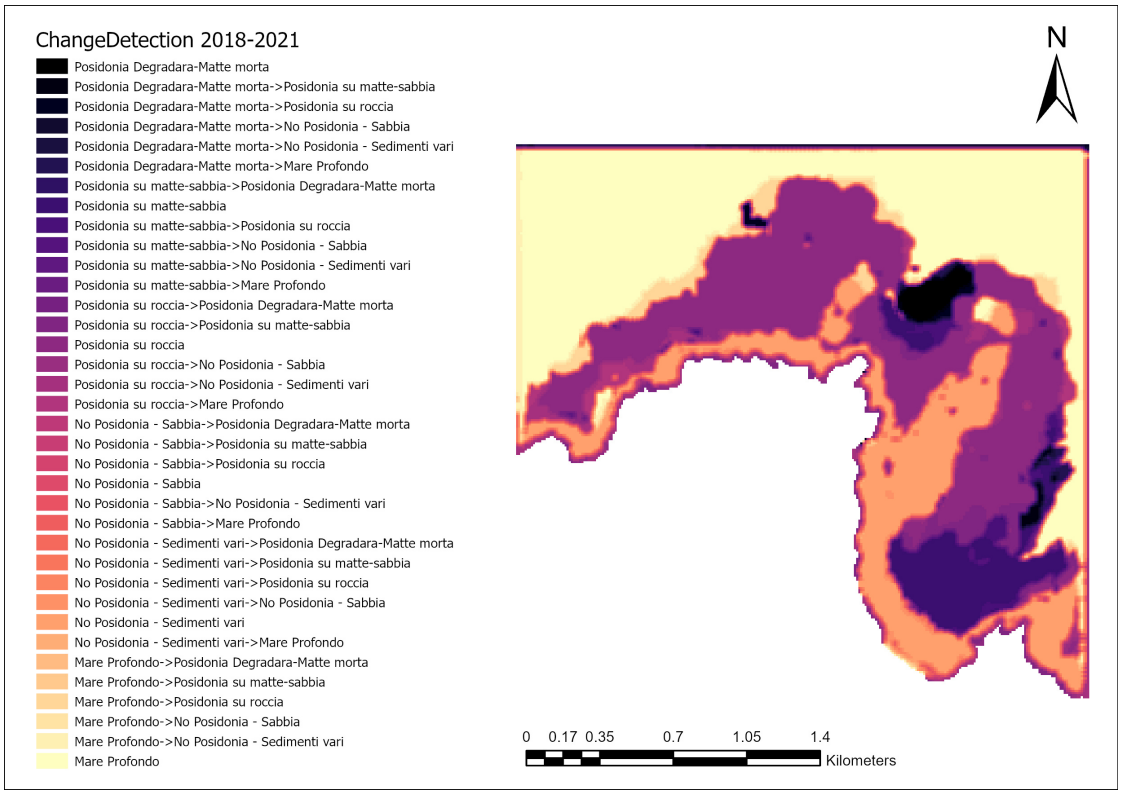


Figure 5.13: Change detection map for 2018–2021.

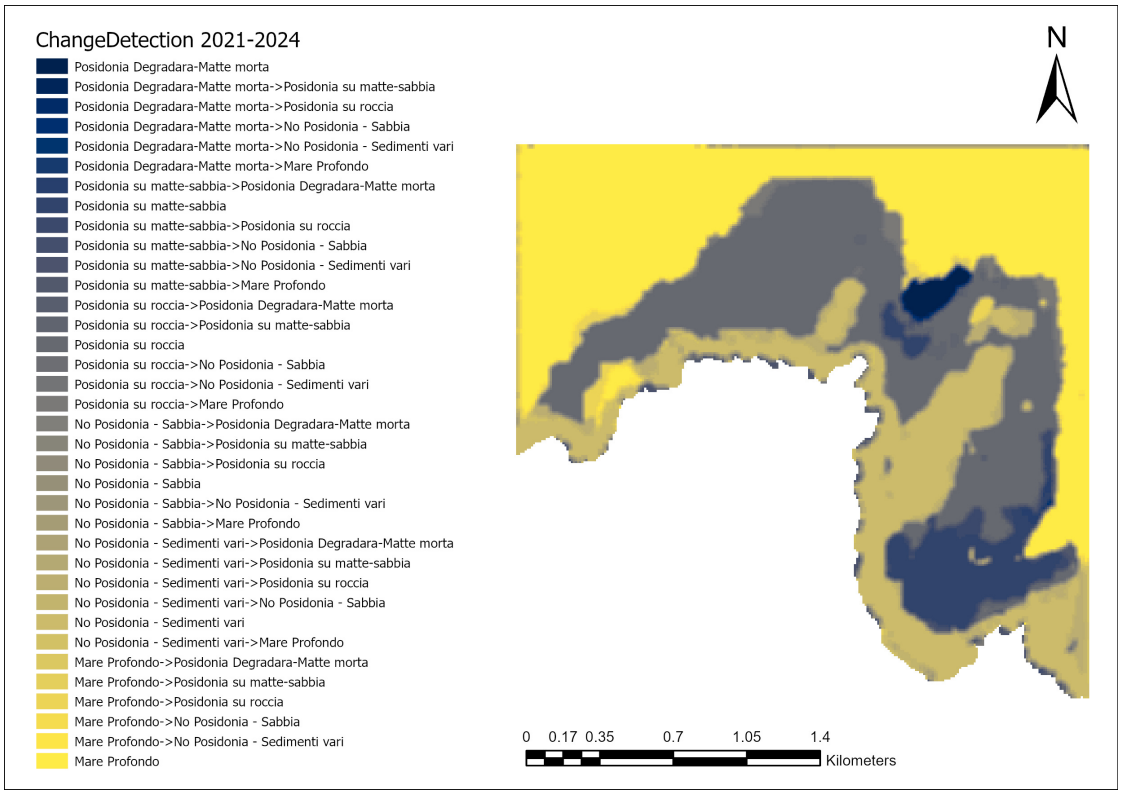


Figure 5.14: Change detection map for 2021–2024.

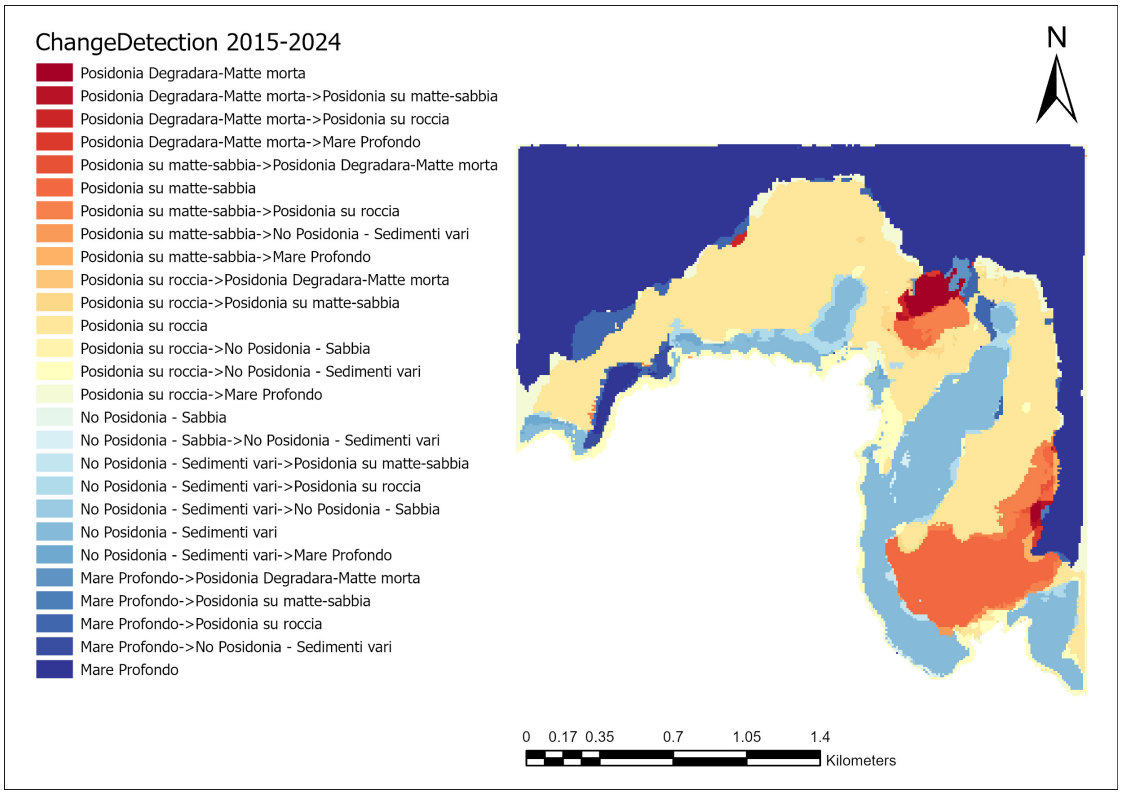


Figure 5.15: Long-term change detection map for 2015–2024.

5.7.3 Numerical assessment of climate change

To quantify long-term changes in benthic habitats, a post-classification comparison was performed between 2015 and 2024 using the *Tabulate Area* tool in ArcGIS Pro. The result is presented as a transition matrix (Table 5.9), where rows represent the reference classes in 2015 and columns the classes in 2024. Diagonal values correspond to stable areas, while off-diagonal values indicate transitions between classes. Both row and column sums are reported to provide the total extent of each class in 2015 and 2024 respectively. Values are expressed in km^2.

Table 5.9: Transition matrix (km^2) showing class changes between 2015 and 2024.

2015 / 2024	Posidonia Degradata	Posidonia Matte sabbia	Posidonia Roccia	No Posidonia Sabbia	No Posidonia Sedimenti vari	Mare Profondo	Row Total (2015)
Posidonia Degradata	0.038	0.005	0.007	0.000	0.000	0.005	0.056
Posidonia Matte-sabbia	0.005	0.270	0.093	0.000	0.007	0.011	0.385
Posidonia Roccia	0.002	0.022	1.209	0.000	0.164	0.173	1.571
No Posidonia Sabbia	0.000	0.000	0.000	0.002	0.007	0.000	0.008
No Posidonia Sedimenti vari	0.000	0.008	0.077	0.001	0.605	0.016	0.707
Mare Profondo	0.013	0.002	0.093	0.000	0.016	1.693	1.817
Column Total (2024)	0.058	0.307	1.479	0.003	0.799	1.898	4.544

Table 5.10: Total class areas (km^2) in 2015 and 2024.

Class	**2015 Area (km^2)**	**2024 Area (km^2)**
Posidonia Degradata	0.056	0.058
Posidonia Matte-sabbia	0.385	0.307
Posidonia Roccia	1.571	1.479
No Posidonia - Sabbia	0.008	0.003
No Posidonia - Sedimenti vari	0.707	0.799
Mare Profondo	1.817	1.898
Total Posidonia (C1-C3)	2.012	1.844
Total Non-Posidonia (C4-C6)	2.532	2.700

The decadal analysis indicates that Posidonia meadows in the study area have undergone a modest but measurable decline, shrinking from just above two square kilometers in 2015 to about 1.84 km^2 in 2024. This corresponds to a loss of roughly 8% over the ten year period. Although small in absolute terms, such a reduction is ecologically significant because Posidonia is one of the slowest growing seagrass species, with expansion typically occurring at only a few centimeters per year. Losses are therefore far more likely to persist and accumulate over time than to be rapidly compensated by natural recovery.

From a climate change perspective, this contraction is consistent with broader expectations for Mediterranean seagrass meadows. Rising sea surface temperatures can impair photosynthetic efficiency and cause thermal stress during marine heatwaves. Increased storminess enhances sediment resuspension and turbidity, reducing underwater light availability and thereby constraining growth at deeper meadow edges. Sea level rise and coastal erosion may further shift depth distributions, forcing meadows to retreat from their lower limits. In parallel, local pressures such as anchoring, dredging, and coastal development compound these climate driven stresses, accelerating regression.

It is important to recognize that part of the observed change may also reflect methodological limitations. Differences in water clarity between satellite acquisitions, edge effects in transitional habitats, and classification uncertainties can all contribute to apparent gains or losses in mapped cover. Nevertheless, the overall trend detected here aligns with the ecological understanding that Posidonia is far more likely to regress under current climate forcing than to expand. In the discussion chapter, these uncertainties will be revisited, but the evidence points toward a gradual weakening of meadow extent in the study area, consistent with regional scale reports of climate related stress on Mediterranean seagrasses.

Chapter 6: Discussion

6.1 Comparative performance across epochs

Across the four epochs, the workflow delivered stable maps of *Posidonia*. Performance rose as the setup matured. The 2024 run was the best: user's accuracy reached 0.8889 (kappa 0.8504), slightly better than 2015 at 0.8825. The 2018 and 2021 results fell between these two and were consistent with the same overall design.

As detailed in the Results chapter, two issues were present and then progressively reduced: perimeter seamlines at the image frame, and errors at true class boundaries. We limited seamlines by using tile overlap and padding and also reduced boundary errors mainly by tuning class weights while keeping enough context with padding. Together these steps produced cleaner maps over time: from 2018 to 2024 the frame stripes almost disappeared and the transitions sharpened, with 2024 showing the cleanest borders and a small accuracy gain over 2015.

6.2 Change detection across epochs

Changes in *Posidonia* usually occur over long time scales. A longer record would give a clearer view of climate and human impacts, but this was not available. To highlight the overall signal with the data at hand, we focused on the widest interval, comparing the first and last epochs (2015 vs. 2024) and did not interpret the shorter intermediate periods.

Using post-classification comparison, the total area of *Posidonia* classes in the study site decreased from $2.012\,\mathrm{km}^2$ in 2015 to $1.844\,\mathrm{km}^2$ in 2024. This is a net loss of $0.168\,\mathrm{km}^2$, which corresponds to about 8.3% of the 2015 extent. As noted in the Results chapter, we reduced false change by masking a thin rim near the image frame, enforcing a minimum mapping unit, and ignoring isolated single-pixel flips.

These figures should be read as an estimate rather than a definitive rate of decline. Uncertainty in change maps is higher than in single-year maps because errors can accumulate from both dates. More confidence would require a longer time series, water-column (depth) correction based on accurate bathymetry, tighter control of

seasonal and illumination differences, and additional ground truth to validate detected gains and losses.

6.3 Hyperparameters and model selection

The training of the first model (2015) took the longest because we started without prior guidance. Once that model was tuned, we reused its configuration as a baseline for later years. This reduced search time and made training more stable.The best results came from a simple and reliable combination of model and training choices. For this task and imagery, the settings below were consistently strong across years.

Table 6.1: Best configuration

	Architecture	Backbone	Batch size	Chip size	Stride	Padding*	Val. split	Oversampled classes
Best config	DeepLabv3	ResNet-34	16	30	15	14	0.20	[1, 2, 3]

* For 2015, a padding of 4 worked better.

Among all the choices, **class weights** were the hardest to get right. With six classes, the search space is large, and the class balance changes with year, water clarity, and substrate. We also saw that small weight changes sometimes could move accuracy by a noticeable margin, especially near edges and in degraded matte. Limited compute meant some trainings took one to two days, so we had to make careful, incremental changes instead of broad sweeps.This meant many trials were needed to reach a stable set of weights for each year.

6.4 Limitations and challenges

- **Spatial resolution.** Sentinel-2 at 10 m works for regional mapping, but it cannot fully resolve very narrow fringes or small fragmented patches.
- **Bathymetry and optics.** Depth and water clarity shape bottom reflectance. Better bathymetry would allow depth normalization and reduce confusion in deeper or turbid waters.
- **Ground truth.** Limited field data constrained model calibration and validation. Underwater photos and photogrammetry are costly and time consuming but provide precise labels. Within the POSEIDON project, researchers of Politecnico di Torino have already collected this type of information at Culuccia island, setting an important precednt for integrating field campaigns with remote sensing workflows (Figure 6.1).

- **Temporal span.** Sentinel-2 starts in 2015, so trend length is short. Adding older archives would help separate short-term variability from long-term change.
- **Compute budget.** Long training times reduced how many model, loss, and schedule options we could try.

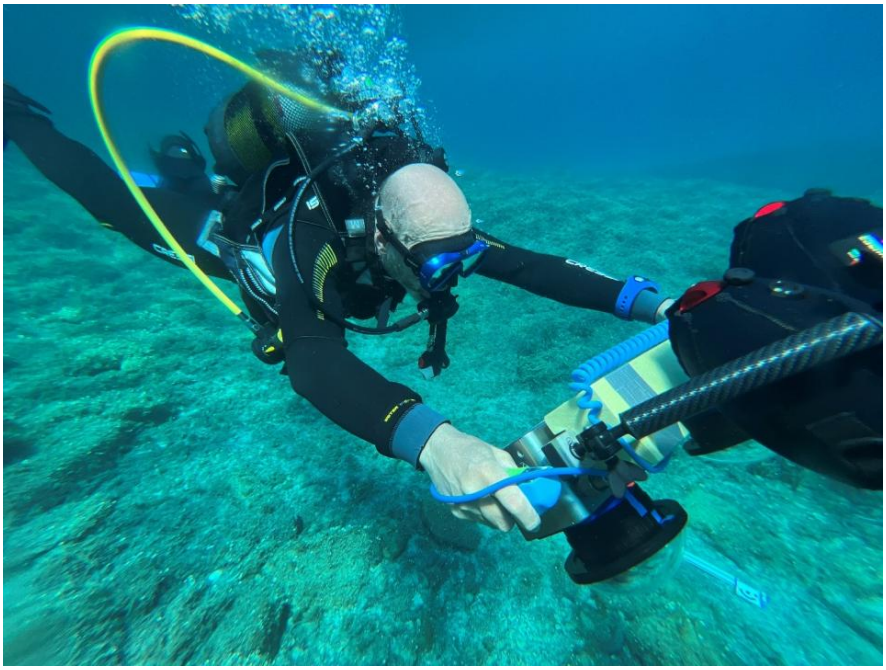


Figure 6.1: Recent field activity of the Politecnico di Torino DIRECT team at Culuccia Island. *Source:* Politecnico di Torino.

6.5 Implications

This workflow is a practical option for regional seagrass monitoring. It is cost-effective, based on open imagery, and repeatable over time, which helps build consistent series for management. The work is aligned with the goals of the POSEIDON initiative on climate impacts to *Posidonia*, and it is consistent with the aims of FAIR under the National Recovery and Resilience Plan.

6.6 Recommendations for future work

- **Sharper imagery.** Add higher-resolution satellites to better capture narrow edges and reduce boundary errors.
- **Depth-aware correction.** Combine multispectral data with accurate bathymetry to normalize for depth and optics, improving class separability.

- **Richer labels.** Plan targeted field campaigns (photos, photogrammetry) to label difficult classes and transitions.
- **Longer timelines.** Bring in pre-2015 archives and cross-check Sentinel-2 with other platforms to strengthen trends.
- **Smarter training.** Use HPC and principled search (e.g., Bayesian optimization) and try imbalance-aware losses (e.g., focal, Dice variants) to reduce manual weight tuning.
- **Transfer and robustness.** Test the recipe on new sites and measure how much re-tuning is needed.

Chapter 7: Conclusions

7.1 Summary of contributions

P.O. meadows in the Mediterranean, widely reported to store more CO_2 per unit area than many forests, are vital to the marine ecosystem yet vulnerable to warming and human pressure. This thesis developed and tested a practical workflow that integrates freely available Sentinel–2 imagery with deep learning to map and monitor *P.O.* meadows. The approach was applied to the Capo Testa Marine Protected Area for 2015–2024. Semantic segmentation models were trained and validated, producing reliable maps of seagrass extent that illustrate both the promise and the current limits of Earth observation for underwater habitat monitoring. By leveraging multispectral and multitemporal data, the study contributes methods and tools that support the development of new techniques for seafloor mapping and, more broadly, for marine conservation.

The results show that a remote sensing pipeline can quantify decadal changes in seagrass cover at regional scale. The method offers a cost–effective and reproducible complement to field campaigns, and it aligns with initiatives that promote non–destructive, georeferenced monitoring of priority habitats.

7.2 Key findings

- Multispectral Sentinel–2 imagery combined with modern semantic segmentation effectively separates seagrass from surrounding substrates.
- Application to a 2015–2024 time series reveals measurable changes in meadow extent at Capo Testa. While modest, these changes are ecologically meaningful and consistent with combined anthropogenic and climate pressures.
- Sentinel–2 remains a useful base for long–term monitoring thanks to its spatial resolution, shallow–water spectral bands, and continuous availability since 2015.
- The workflow is likely usable in other parts of the island, as long as the bottom types (sand vs. rock), the seagrass structure and density, the depth range, and

the water optics (clarity, color, sun–glint) are similar to those in the training area. If these factors change noticeably, a small amount of re-tuning (for example, class weights or a short fine-tuning pass) may be needed to keep accuracy.

7.3 Closing remark

Open satellite data combined with a focused deep learning workflow can track changes in *Posidonia oceanica* at regional scale with useful accuracy. This is not a replacement for field surveys, but it helps target where and when to sample, reducing costs and effort. With modest additions such as better bathymetry, a bit more ground truth, and stronger computing, the same approach can be extended to nearby coasts that have similar water optics and seabed types. Used this way, it can support routine, evidence-based monitoring for conservation and coastal planning. A multidisciplinary view that links geomatics, biology, and ecology will remain essential for interpreting maps and connecting patterns with their drivers.

Acknowledgements

This thesis was carried out in the Geomatics Lab of the DIATI Department at Politecnico di Torino. I am grateful for the supportive environment, the access to instruments and software, and the many practical discussions that helped me turn ideas into a working workflow. I am especially thankful to my co-supervisors, Stefania Manca and Francesca Gallitto, for their practical insight, timely feedback, and steady encouragement at every step.

From June 17 to 21, 2024, as part of the DIRECT Team Activities, I joined a data acquisition campaign on Culuccia Island, Sardinia, together with researchers and professors. The goal was to collect geospatial and environmental data using terrestrial, aerial, and underwater survey techniques to monitor the island's natural and marine environment. That hands-on experience motivated me to pursue this project, whose aims are closely aligned with the Culuccia campaign. I would like to thank all DIRECT group members and researchers for that invaluable experience.

As noted elsewhere in this thesis, the study aligns with the broader objectives of the POSEIDON project, particularly the investigation of climate change impacts on Posidonia. The artificial intelligence and deep learning workflow developed here is aligned and consistent with the objectives of the FAIR Foundation under the National Recovery and Resilience Plan.

To my friends, thank you for standing by me, sharing your time, and making the difficult parts lighter. To my family, thank you for your unwavering support across the distance, for your patience, and for believing in me at every step. Your encouragement made this work possible.

Bibliography

Books

Hemminga, M. A. and C. M. Duarte (2000). *Seagrass Ecology*. Cambridge University Press.

Huete, A.R. (2004). *Remote Sensing for Environmental Monitoring and Characterization*. Elsevier.

Book chapters

Boudouresque, C., G. Bernard, G. Pergent, A. Shili, and M. Verlaque (2006). "Regression of Mediterranean seagrasses caused by natural processes and anthropogenic disturbances and stress: a critical review". In: *Seagrasses: Biology, Ecology and Conservation*. Ed. by A. W. D. Larkum, R. J. Orth, and C. M. Duarte. Springer, pp. 151–178.

Danovaro, R. and F. Boero (2019). "Italian Seas". In: *World Seas: An Environmental Evaluation, Volume I: Europe, the Americas and West Africa*. 2nd ed. Elsevier, pp. 283–306. DOI: 10.1016/B978-0-12-805068-2.00044-9.

Pakela, Julia and Issam El Naqa (2022). "Overview of Deep Machine Learning Methods". In: *Machine and Deep Learning in Oncology, Medical Physics and Radiology*. Springer International Publishing, pp. 51–77. DOI: 10.1007/978-3-030-83047-2_4. URL: https://doi.org/10.1007/978-3-030-83047-2_4.

Journal articles

Alastal, Abdelhalek I. and Ashraf Hassan Shaqfa (2022). "GeoAI Technologies and Their Application Areas in Urban Planning and Development: Concepts, Opportunities and Challenges in Smart City (Kuwait, Study Case)". In: *Journal of Data Analysis and Information Processing* 10.2, pp. 65–84. DOI: 10.4236/jdaip.2022.102007. URL: https://www.scirp.org/journal/paperinformation?paperid=116308.

Anonymous (2020). “A Modified U-net-based Architecture for Segmentation of Satellite Images”. In: *International Journal of Advanced Computer Science and Applications*. Adapted figure used in this thesis.

— (2024). “AI-driven remote sensing enhances Mediterranean seagrass mapping”. In: *Scientific Reports* 14.59091. DOI: 10.1038/s41598-024-59091-7. URL: https://www.nature.com/articles/s41598-024-59091-7.

Armeniakos, Giorgos, Georgios Zervakis, Dimitrios Soudris, and Jörg Henkel (2022). “Hardware Approximate Techniques for Deep Neural Network Accelerators: A Survey”. In: *ACM Computing Surveys* 55.4, pp. 1–36. DOI: 10.1145/3527156. URL: https://doi.org/10.1145/3527156.

Bernardis, M. et al. (2023). “Use of ICESat-2 and Sentinel-2 Open Data for the Derivation of Bathymetry in Shallow Waters: Case Studies in Sardinia and in the Venice Lagoon”. In: *Remote Sensing* 15.11, p. 2944. DOI: 10.3390/rs15112944. URL: https://www.mdpi.com/2072-4292/15/11/2944.

Borfecchia, F. et al. (2019). “Landsat 8 OLI satellite data for mapping of the Posidonia oceanica and benthic habitats of coastal ecosystems”. In: *International Journal of Remote Sensing* 40.5, pp. 1548–1575. DOI: 10.1080/01431161.2018.1528408.

Boudouresque, C., G. Bernard, G. Pergent, A. Shili, and M. Verlaque (2009). “Regression of Mediterranean seagrasses caused by natural processes and anthropogenic disturbances: a critical review”. In: *Botanica Marina* 52.5, pp. 395–418.

Boudouresque, C. et al. (2017). “The high heritage value of the Mediterranean sandy beaches, with a particular focus on the Posidonia oceanica “banquettes”: A review”. In: *Scientific Reports of Port-Cros National Park* 31, pp. 23–70.

Buia, Maria C., Maria C. Gambi, and V. Zupo (2004). “Structure and functioning of Mediterranean seagrass ecosystems: an overview”. In: *Biology of Seagrasses*, pp. 167–190.

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: 10.1016/j.ecoinf.2020.101053.

Campillo-Tamarit, Noelia, Juan V. Molner, and Juan M. Soria (2025). “Remote Sensing Tools for Monitoring Marine Phanerogams: A Review of Sentinel and Landsat Applications”. In: *Journal of Marine Science and Engineering* 13.2, p. 292. DOI: 10.3390/jmse13020292.

Chowdhury, Masuma et al. (2024). “AI-driven remote sensing enhances Mediterranean seagrass monitoring and conservation to combat climate change and anthropogenic impacts”. In: *Scientific Reports* 14, p. 8360. DOI: 10.1038/s41598-024-59091-7.

Cingano, Paolo et al. (2024). "Seagrasses on the move: Tracing the multi-decadal species distribution trends in lagoon meadows using Landsat imagery". In: *Ecological Informatics* 82, p. 102685. DOI: 10.1016/j.ecoinf.2024.102685.

Davies, Bede Ffinian Rowe et al. (2024). "Intertidal seagrass extent from Sentinel-2 time-series show distinct trajectories in Western Europe". In: *Remote Sensing of Environment* 312, p. 114340. DOI: 10.1016/j.rse.2024.114340.

Francour, P. et al. (1999). "Effects of boat anchoring in Posidonia oceanica seagrass beds in the Port-Cros National Park (north-western Mediterranean Sea)". In: *Aquatic Conservation: Marine and Freshwater Ecosystems* 9.4, pp. 391–400. DOI: 10.1002/(SICI)1099-0755(199907/08)9:4<391::AID-AQC357>3.0.CO;2-8.

Giménez-Romero, Alex, Dhafer Ferchichi, Pablo Moreno-Spiegelberg, Tomàs Sintes, and Manuel A. Matías (2024). "Mapping the distribution of seagrass meadows from space with deep convolutional neural networks". In: *bioRxiv*. DOI: 10.1101/2024.03.21.586047.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

Kim, H. et al. (2023). "Comparison of Satellite Imagery for Identifying Seagrass Distribution: GeoEye-1, Sentinel-2 MSI, and Landsat-8 OLI". In: *Journal of Marine Science and Engineering* 11.4, p. 701. DOI: 10.3390/jmse11040701.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2017). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Communications of the ACM* 60.6, pp. 84–90. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

Lopez y Royo, Carlos L., Giulia Casazza, Christine Pergent-Martini, and Gérard Pergent (2010). "A biotic index using Posidonia oceanica (BiPo) to evaluate ecological status of coastal waters". In: *Ecological Indicators* 10.2, pp. 380–389. DOI: 10.1016/j.ecolind.2009.04.016.

Maurya, Abhishek, Akashdeep, Payal Mittal, and Rohit Kumar (2023). "A modified U-net-based architecture for segmentation of satellite images on a novel dataset". In: *Ecological Informatics* 75, p. 102078. DOI: 10.1016/j.ecoinf.2023.102078.

Maxwell, Aaron E., Timothy A. Warner, and Fang Fang (2019). "Large-Area, High Spatial Resolution Land Cover Mapping Using Random Forests, GEOBIA, and NAIP Orthophotography". In: *Remote Sensing* 11.23, p. 2906. DOI: 10.3390/rs11232906. URL: https://doi.org/10.3390/rs11232906.

Mederos-Barrera, Antonio, Javier Marcello, Francisco Eugenio, and Enrique Hernández (2022). "Seagrass mapping using high resolution multispectral satellite imagery: A comparison of water column correction models". In: *International Journal of*

Applied Earth Observation and Geoinformation 113, p. 102990. DOI: 10.1016/j.jag.2022.102990.

Milazzo, Massimo, Fabio Badalamenti, Giulia Ceccherelli, and Renato Chemello (2004). "Boat anchoring on Posidonia oceanica beds in a marine protected area (Italy, western Mediterranean): effect of anchor types in different anchoring stages". In: *Journal of Experimental Marine Biology and Ecology* 299.1, pp. 51–62. DOI: 10.1016/j.jembe.2003.08.009.

Montefalcone, Monica (2009). "Ecosystem health assessment using the Mediterranean seagrass *Posidonia oceanica*: A review". In: *Ecological Indicators* 9.1, pp. 93–104. DOI: 10.1016/j.ecolind.2008.09.013. URL: https://www.researchgate.net/publication/222517239_Ecosystem_health_assessment_using_the_Mediterranean_seagrass_Posidonia_oceanica_A_review.

Peng, Jianghai, Jiwei Li, Thomas C. Ingalls, Steven R. Schill, Hannah R. Kerner, and Gregory P. Asner (2025). "A novel deep learning algorithm for broad scale seagrass extent mapping in shallow coastal environments". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 220, pp. 277–294. DOI: 10.1016/j.isprsjprs.2024.12.008.

Pergent, G., J. Romero, C. Pergent-Martini, M. A. Mateo, and C. F. Boudouresque (1994). "Primary production, stocks and fluxes in Posidonia oceanica beds". In: *Marine Ecology Progress Series* 106, pp. 139–146.

Poursanidis, Dimitris, Konstantinos Topouzelis, and Nektarios Chrysoulakis (2018). "Mapping coastal marine habitats and delineating the deep limits of the Neptune's seagrass meadows using very high resolution Earth observation data". In: *International Journal of Remote Sensing* 39.23, pp. 8694–8717. DOI: 10.1080/01431161.2018.1490974.

Ramachandran, G. and S. Kannan (Nov. 2021). "Artificial Intelligence and Deep Learning Applications: A Review". In: *Journal of Artificial Intelligence, Machine Learning and Neural Network* 12, pp. 10–13. DOI: 10.55529/jaimlnn.12.10.13. URL: https://doi.org/10.55529/jaimlnn.12.10.13.

Rende, Sante Francesco et al. (2020). "Ultra-High-Resolution Mapping of *Posidonia oceanica* (L.) Delile Meadows through Acoustic, Optical Data and Object-based Image Classification". In: *Journal of Marine Science and Engineering* 8.9, p. 647. DOI: 10.3390/jmse8090647.

Sagawa, Tatsuyuki et al. (2010). "Using bottom surface reflectance to map coastal marine areas: A new application of Lyzenga's model". In: *International Journal of Remote Sensing* 31.12, pp. 3051–3064. DOI: 10.1080/01431160903154341.

Telesca, L. et al. (2015). "Seagrass meadows (Posidonia oceanica) distribution and trajectories of change". In: *Scientific Reports* 5, p. 12505. DOI: 10.1038/srep12505.

Topouzelis, K., D. Makri, N. Stoupas, A. Papakonstantinou, and S. Katsanevakis (2018). "Seagrass mapping in Greek territorial waters using Landsat-8 satellite images". In: *International Journal of Applied Earth Observation and Geoinformation* 67, pp. 98–113. DOI: 10.1016/j.jag.2017.12.012.

Traganos, D. and P. Reinartz (2018). "Mapping Mediterranean seagrasses with Sentinel-2 imagery". In: *Marine Pollution Bulletin* 134, pp. 197–209. DOI: 10.1016/j.marpolbul.2017.09.036.

Traganos, Dimosthenis, Bharat Aggarwal, Dimitris Poursanidis, Konstantinos Topouzelis, Nektarios Chrysoulakis, and Peter Reinartz (2018). "Towards Global-Scale Seagrass Mapping and Monitoring Using Sentinel-2 on Google Earth Engine: The Case Study of the Aegean and Ionian Seas". In: *Remote Sensing* 10.8, p. 1227. DOI: 10.3390/rs10081227.

Watson, Cody, Nathan Cooper, David Nader Palacio, Kevin Moran, and Denys Poshyvanyk (2022). "A Systematic Literature Review on the Use of Deep Learning in Software Engineering Research". In: *ACM Transactions on Software Engineering and Methodology* 31.2, pp. 1–58. DOI: 10.1145/3485275. URL: https://doi.org/10.1145/3485275.

Waycott, M. et al. (2009). "Accelerating loss of seagrasses across the globe threatens coastal ecosystems". In: *Proceedings of the National Academy of Sciences* 106.30, pp. 12377–12381. DOI: 10.1073/pnas.0905620106.

Conference papers

Ceccherelli, Giulia et al. (2024). "Multitemporal seagrass mapping and monitoring of Posidonia meadows and banquettes for blue carbon conservation: the POSEIDON Project". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XLVIII-1, pp. 65–72. DOI: 10.5194/isprs-archives-XLVIII-1-2024-65-2024.

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille (2017a). "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 40. 4, pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184.

Chen, Liang-Chieh, George Papandreou, Florian Schroff, and Hartwig Adam (2017b). "Rethinking Atrous Convolution for Semantic Image Segmentation". In: *arXiv preprint arXiv:1706.05587*. URL: https://arxiv.org/abs/1706.05587.

Dattola, L. et al. (2018). "Comparison of Sentinel-2 and Landsat-8 OLI satellite images vs. high spatial resolution images (MIVIS and WorldView-2) for mapping Posidonia oceanica meadows". In: *Remote Sensing of the Ocean, Sea Ice, Coastal Waters,*

and Large Water Regions 2018. Vol. 10784. SPIE, pp. 252–262. DOI: 10.1117/12.2325757.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015a). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

— (2015b). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, pp. 234–241. URL: https://arxiv.org/abs/1505.04597.

Scarpetta, Marco, Paolo Affuso, Maddalena de Virgilio, Maurizio Spadavecchia, Gregorio Andria, and Nicola Giaquinto (2022). "Monitoring of Seagrass Meadows Using Satellite Images and U-Net Convolutional Neural Network". In: *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6. DOI: 10.1109/I2MTC48687.2022.9806535.

Tappert, Charles C. (2019). "Who Is the Father of Deep Learning?" In: *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, pp. 343–348. DOI: 10.1109/csci49370.2019.00067. URL: https://doi.org/10.1109/csci49370.2019.00067.

Datasets and portals

Berríos, Ignacio T. (2020). *DeepLabv3 Explained for Semantic Segmentation*. Medium. Accessed: 2025-08-17. URL: https://medium.com/@itberrios6/deeplabv3-c0c8c93d25a4.

Capo Testa – Punta Falcone MPA (2024). *Official Website of the Marine Protected Area*. Accessed on August 5, 2025. URL: https://areamarinaprotettacapotestapuntafalcone.it/contenuti/243659/organizzazione.

Code, Papers with (2023). *ResNet: Residual Networks Explained*. Accessed: 2025-08-18. URL: https://paperswithcode.com/method/resnet.

Consiglio Nazionale delle Ricerche (CNR) (2025). *Shapefile dataset for No Posidonia – sabbia*. Accessed on August 7, 2025. URL: http://libeccio.bo.ismar.cnr.it:8080/geonetwork/pnrr/api/records/115e1ddb-0fc5-4b09-a437-ba64f5d0eaff.

Copernicus Data Space Ecosystem (2025). *Sentinel-2 Mission Overview and Data Access*. Accessed on August 6, 2025. URL: https://dataspace.copernicus.eu/explore-data/data-collections/sentinel-data/sentinel-2.

Damiana (2023). *Posidonia oceanica: alga o pianta?* Accessed: 2025-09-06. URL: https://www.lustricadiving.it/posidonia-oceanica-alga-o-pianta/.

Datature (2023). *A Guide to Using DeepLabv3 for Semantic Segmentation*. Accessed: 2025-08-17. URL: https://datature.io/blog/a-guide-to-using-deeplabv3-for-semantic-segmentation.

ESA Sentinel Wiki (2025). *Sentinel-2 Product Format and Folder Structure*. Accessed on August 6, 2025. URL: https://sentiwiki.copernicus.eu/web/s2-products.

ESRI (2024). *Spectral Profile Chart*. https://pro.arcgis.com/en/pro-app/3.3/help/analysis/geoprocessing/charts/spectral-profile-chart.htm. Accessed: 20 August 2025.

— (2025a). *Change Detection Wizard (Image Analyst)*. https://pro.arcgis.com/en/pro-app/latest/help/analysis/image-analyst/the-change-detection-wizard.htm. Accessed: 2 September 2025.

— (2025b). *Create Accuracy Assessment Points (Spatial Analyst)*. https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/create-accuracy-assessment-points.htm. Accessed: 27 August 2025.

Esri (2024). *Classify Pixels Using Deep Learning (Image Analyst)*. https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/classify-pixels-using-deep-learning.htm. Accessed 20 Aug 2025.

— (2025a). *Resample (Data Management)—ArcGIS Pro Tool Reference*. Accessed on August 6, 2025. URL: https://pro.arcgis.com/en/pro-app/3.3/tool-reference/data-management/resample.htm.

— (2025b). *Training Samples Manager—ArcGIS Pro Documentation*. Accessed on August 7, 2025. URL: https://pro.arcgis.com/en/pro-app/latest/help/analysis/image-analyst/training-samples-manager.htm.

Esri Developer (2023). *Deep Learning in ArcGIS Pro*. Accessed: 2025-08-17. URL: https://pro.arcgis.com/en/pro-app/latest/help/analysis/deep-learning/deep-learning-in-arcgis-pro.htm.

European Marine Observation and Data Network (EMODnet) (2025). *Map of the Week: Posidonia oceanica Distribution*. Accessed on August 7, 2025. URL: https://emodnet.ec.europa.eu/en/map-week-posidonia-oceanica-distribution.

European Space Agency (2025a). *Sentinel-2 Level-2A Surface Reflectance Product Description*. Accessed on August 6, 2025. URL: https://documentation.dataspace.copernicus.eu/Data/SentinelMissions/Sentinel2.html#sentinel-2-level-2a-surface-reflectance.

— (2025b). *Sentinel-2 Scene Classification (SCL) Description*. Accessed on August 7, 2025. URL: https://documentation.dataspace.copernicus.eu/Data/SentinelMissions/Sentinel2.html.

GeeksforGeeks (2023a). *DeepLab v3 — Architecture and Working Explained*. Accessed: 2025-08-17. URL: https://www.geeksforgeeks.org/deeplab-v3-architecture-and-working-explained/.

GeeksforGeeks (2023b). *U-Net Architecture for Image Segmentation*. Accessed: 2025-08-17. URL: https://www.geeksforgeeks.org/u-net-architecture-explained/.

— (2025). *CNN — Introduction to Padding*. https://www.geeksforgeeks.org/machine-learning/cnn-introduction-to-padding/. Accessed: 22 August 2025.

ISPRA and partners (2025). *Marine Geospatial Information Coastal Database (MaGIC)*. Accessed on August 7, 2025. URL: https://github.com/pcm-dpc/MaGIC.

Jain, Abhishek (2019). *All About Convolutions, Kernels and Features in CNN*. Medium. Accessed: 2025-08-17. URL: https://medium.com/@abhishekjainindore24/all-about-convolutions-kernels-features-in-cnn-c656616390a1.

Mallick, Satya (2016). *Understanding Convolutional Neural Networks (CNN)*. LearnOpenCV. Accessed: 2025-08-17. URL: https://learnopencv.com/understanding-convolutional-neural-networks-cnn/.

Monks, AI (2023). *Understanding Sentinel-2 L2A Scene Classification Map with Python Codes*. Accessed on August 7, 2025. URL: https://medium.com/aimonks/understanding-sentinel-2-l2a-scene-classification-map-with-python-codes-5973938c95d4.

Regione Autonoma della Sardegna (2025). *Sardegna SIRA Portal – Marine Protected Areas*. Accessed on August 5, 2025. URL: https://portal.sardegnasira.it/aree-marine-protette.

Chapter A: Appendix

```
1
2
3  # Calculate Weights
4
5  import os
6  import numpy as np
7  import rasterio
8  from collections import Counter
9   # Set your label folder path
10 labels_folder = r"F:\Payam\21_05_Poseidon\N2\export\TS2\TS2Tile10\labels"
11  label_arrays = []
12  # Load all label rasters into memory
13 for filename in os.listdir(labels_folder):
14     if filename.endswith(".tif"):
15         path = os.path.join(labels_folder, filename)
16         with rasterio.open(path) as src:
17             label = src.read(1)
18             label_arrays.append(label)
19  # Flatten and count class frequencies
20 all_labels = np.concatenate([arr.flatten() for arr in label_arrays])
21 class_counts = dict(Counter(all_labels))
22  # Calculate pixel counts and class weights for class IDs 1 to 6
23 pixel_counts = []
24 print("Class Distribution (only among labeled pixels):")
25 for class_id in range(1, 7):
26     count = class_counts.get(class_id, 0)
27     pixel_counts.append(count)
28     percentage = (count / len(all_labels)) * 100 if len(all_labels) > 0
       ↪  else 0
29     print(f"Class {class_id}: {count} pixels ({percentage:.2f}%)")
30  # Compute inverse-frequency class weights
31 total_labeled_pixels = sum(pixel_counts)
```

```
num_classes = len(pixel_counts)
 class_weights = [
    total_labeled_pixels / (num_classes * count) if count > 0 else 0
    for count in pixel_counts
]
 # Optionally normalize weights (so they sum to number of classes)
normalize = False
if normalize:
    weight_sum = sum(class_weights)
    class_weights = [w * num_classes / weight_sum for w in class_weights]
 # Output final weights
print("\n✓ Final Class Weights:")
for i, weight in enumerate(class_weights, start=1):
    print(f"Class {i}: {weight:.4f}")


# Train Deep Learning Model

# Import necessary libraries
from arcgis.learn import UnetClassifier, prepare_data
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report,
↪  precision_recall_fscore_support
import pandas as pd
import numpy as np
import torch
import os
import time
# Start timing
start_time = time.time()
# Define paths
training_data_path =
↪  r"F:\Payam\11.09.2018\CapoTesta2_11.09.2018\Export\Tile30"
model_save_path =
↪  r"F:\Payam\11.09.2018\CapoTesta2_11.09.2018\Result\T30DeeplabW1"
# Step 1: Define Optimized Class Weights
class_weights = torch.tensor([w1, w2, w3, w4, w5, w6], dtype=torch.float32)
# Step 2: Define Class Mapping
class_map = {
    1: "Posidonia Degradara-Matte morta",
    2: "Posidonia su matte-sabbia",
```

```
    3: "Posidonia su roccia",
    4: "No Posidonia - Sabbia",
    5: "No Posidonia - Sedimenti vari",
    6: "Mare Profondo",
}
print("Class Mapping (Excluding Class 0):")
for class_id, class_name in class_map.items():
    print(f"Class {class_id}: {class_name}")
# Step 3: Prepare Data with Balanced Oversampling
data = prepare_data(
    training_data_path,
    batch_size=16,
    chip_size=30,
    val_split_pct=0.2,
    class_mapping=class_map,
    class_balancing=True,
    class_weights=class_weights.tolist(),
    oversample=True,
    oversample_classes=[1, 2, 3]
)
# Step 4: Initialize the Model
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")
model = UnetClassifier(data, model='deeplabv3')
model.learn.model.to(device)
# Training
train_losses = []
epoch_train_losses = []
valid_losses = []
accuracies = []
dice_scores = []
print("Starting training...")
lr_schedule = [(1400, 4e-4), (1100, 3e-4), (800, 2e-4), (600, 1e-4), (400,
↪  5e-5)]
total_epochs = 0
for epochs, lr in lr_schedule:
    model.fit(epochs=epochs, lr=lr)
    recorder = model.learn.recorder
    train_losses.extend(recorder.losses)
    valid_losses.extend(recorder.val_losses[-epochs:])
    for metric_tuple in recorder.metrics[-epochs:]:
```

```
        accuracies.append(metric_tuple[0])
        dice_scores.append(metric_tuple[1])
    batches_per_epoch = len(recorder.losses) // (total_epochs + epochs)
    for e in range(total_epochs, total_epochs + epochs):
        start_idx = e * batches_per_epoch
        end_idx = (e + 1) * batches_per_epoch

        ↪ epoch_train_losses.append(np.mean(recorder.losses[start_idx:end_idx]))
    total_epochs += epochs
print("Training complete!")
# Step 6: Evaluation with Corrected Label Mapping
print("Evaluating model performance...")
y_true, y_pred = [], []
model.learn.model.eval()
with torch.no_grad():
    for xb, yb in data.valid_dl:
        xb = xb.to(device)
        yb = yb.to(device)
        preds = model.learn.model(xb)
        preds = torch.argmax(preds, dim=1).cpu().numpy()
        yb = yb.cpu().numpy().squeeze(1)
        mask = yb > 0
        mapped_preds = preds[mask] + 1  # Shift predicted classes to match
        ↪ label IDs (1{6)
        y_true.extend(yb[mask].flatten())
        y_pred.extend(mapped_preds.flatten())
# Compute Metrics
labels = list(class_map.keys())
label_names = [class_map[i] for i in labels]
precision, recall, f1, _ = precision_recall_fscore_support(
    y_true, y_pred, labels=labels, zero_division=0
)
metrics_df = pd.DataFrame({
    "Class": label_names,
    "Precision": precision,
    "Recall": recall,
    "F1-Score": f1
})
print("\nModel Performance Metrics:")
print(metrics_df)
output_path = os.path.join(model_save_path, "model_metrics_final.html")
```

```
metrics_df.to_html(output_path, index=False)
print(f"Metrics saved to: {output_path}")
# Step 7: Save Model
os.makedirs(model_save_path, exist_ok=True)
model.learn.model.cpu()
torch.save(model.learn.model.state_dict(), os.path.join(model_save_path,
↪  "model_final.pth"))
print(f"Model saved at: {model_save_path}/model_final.pth")
model.save(model_save_path, compute_metrics=False,
↪  save_inference_file=True)
# Step 8: Graphs
print("Creating training graphs...")
epochs_x = range(1, len(valid_losses) + 1)
plt.figure(figsize=(10, 6))
plt.plot(epochs_x, epoch_train_losses, label='Training Loss (avg per
↪  epoch)')
plt.title('Training Loss Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(model_save_path, 'training_loss.png'))
plt.close()
plt.figure(figsize=(10, 6))
plt.plot(epochs_x, valid_losses, label='Validation Loss', color='orange')
plt.title('Validation Loss Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(model_save_path, 'validation_loss.png'))
plt.close()
plt.figure(figsize=(10, 6))
plt.plot(epochs_x, accuracies, label='Accuracy', color='green')
plt.title('Accuracy Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(model_save_path, 'accuracy.png'))
plt.close()
```

```
plt.figure(figsize=(10, 6))
plt.plot(epochs_x, dice_scores, label='Dice Score', color='red')
plt.title('Dice Score Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Dice Score')
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(model_save_path, 'dice_score.png'))
plt.close()
print("All training graphs saved successfully!")
elapsed_time = time.time() - start_time
print(f"Elapsed time: {elapsed_time:.2f} seconds.")

```