ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Αναγνώριση αλλαγών της ακτογραμμής και της στάθμης της θάλασσας από δορυφορικές εικόνες με χρήση μηχανικής μάθησης

ΔΙΟΝΥΣΙΟΣ ΓΙΑΝΝΑΡΟΠΟΥΛΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κολλομβάτσος Κωνσταντίνος
Επίκουρος Καθηγητής

Λαμία .... έτος 2024

UNIVERSITY OF THESSALY

Shoreline and Sea Level Changes Recognition from Satellite Images Using Machine Learning

DIONYSIOS GIANNAROPOULOS

FINAL THESIS

ADVISOR

Konstantinos Kolomvatsos
Assistant Professor

Lamia .... year 2024

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις (1), που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. *Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί* ***χωρίς να τα περικλείω σε εισαγωγικά*** *και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

2. *Δέχομαι ότι η αυτολεξεί* ***παράθεση χωρίς εισαγωγικά****, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. *Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια*

4. *Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.*

Ημερομηνία: ……/…../20……

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Περίληψη

Η παρούσα διατριβή ασχολείται με τον υπολογισμό της διείσδυσης της θάλασσας στην ξηρά με την πάροδο του χρόνου με την βοήθεια μεθόδων Μηχανικής Μάθησης. Σύμφωνα με την ΕΕΑ (European Environment Agency, 2024), λόγω της υπερθέρμανσης του πλανήτη η παγκόσμια μέση στάθμη της θάλασσας (Global Mean Sea Level, GMSL) τα τελευταία 200 χρόνια αυξάνεται με μεγαλύτερη ταχύτητα σε σύγκριση με τις προηγούμενες χιλιετίες. Η ποιότητα ζωής του ανθρώπου συνδέεται άμεσα με την στάθμη της θάλασσας, καθώς οι αυξήσεις της GMSL έχουν σοβαρές επιπτώσεις στις φυσικές καταστροφές, όπως πλημμύρες και καταιγίδες. Στο παρελθόν, ολόκληροι πολιτισμοί έχουν εξαφανιστεί εξαιτίας της ανόδου της στάθμης της θάλασσας. Οι διακυμάνσεις της ακτογραμμής ως αποτέλεσμα της ανόδου της στάθμης της θάλασσας μπορούν να αλλάξουν δραματικά το τοπίο, προκαλώντας διάβρωση των ακτών και απώλεια γης, γεγονός που απειλεί την ανθρώπινη ζωή και τις περιουσίες, δημιουργώντας κίνδυνο για τις παράκτιες κοινότητες και υποδομές. Η μεταβαλλόμενη γεωμετρία μιας ακτογραμμής μπορεί να κατανοηθεί καλά μέσω περιοδικών φυσικών ερευνών ή με τηλεπισκόπηση μέσω δορυφόρων. Η δορυφορική τηλεπισκόπηση αποτελεί μια αποτελεσματική και αποδοτική μέθοδο για την παρακολούθηση των διακυμάνσεων της ακτογραμμής.

Ο σκοπός αυτής της πτυχιακής εργασίας είναι να αναπτυχθεί ένα μοντέλο μηχανικής μάθησης που θα μπορεί να αναγνωρίζει και να παρακολουθεί τις αλλαγές στις ακτογραμμές με την πάροδο του χρόνου μέσω τηλεπισκόπησης, επιτυγχάνοντας ακρίβεια συγκρίσιμη με αυτή των δορυφορικών εικόνων. Αυτό το μοντέλο στοχεύει να παρέχει ακριβή ανίχνευση υδάτινων και χερσαίων σωμάτων για ακριβείς υπολογισμούς των αλλαγών της ακτογραμμής, ιδιαίτερα σε περιπτώσεις όπου οι δορυφόροι αντιμετωπίζουν προκλήσεις. Η επιτυχία αυτού του μοντέλου θα συμβάλει στην καλύτερη κατανόηση της δυναμικής των παράκτιων περιοχών, βοηθώντας στην παρακολούθηση του περιβάλλοντος.

Abstract

This thesis focuses on calculating the intrusion of the sea onto land over time using Machine Learning methods. According to EEA (European Environment Agency, 2024), due to global warming, the global mean sea level (GMSL) has been rising at a faster rate over the past 200 years compared to previous millennia. Human quality of life is directly linked to sea level, as increases in GMSL have significant impacts on natural disasters such as floods and storms. In the past, entire civilizations have vanished due to rising sea levels. Changes in coastlines resulting from rising sea levels can dramatically alter landscapes, causing coastal erosion and land loss, posing threats to human life and property, and endangering coastal communities and infrastructure. The changing geometry of a coastline can be well understood through periodic physical surveys or satellite remote sensing, which is an effective and efficient method for monitoring coastline fluctuations.

The purpose of this Thesis is to develop a machine-learning model capable of recognizing and monitoring changes in coastlines over time through remote sensing, achieving accuracy comparable to satellite imagery. This model aims to provide precise detection of water and land bodies for precise calculations of coastline changes, particularly in cases where satellites encounter challenges. The success of this model will enhance understanding of the dynamics of coastal regions, supporting environmental monitoring efforts.

Contents

Introduction

Approximately two-thirds of the world's population resides in coastal areas, which hold significant economic and social importance. Proper management of these areas is crucial to preserve and enhance their value for future generations. Beaches, stretching along all coastlines, are inherently dynamic, with sand constantly being redistributed by waves, wind, and coastal currents (Pianca, Holman, & Siegle, 2015).

This thesis introduces an approach to delineating water bodies and land boundaries and calculating coastal changes using high-resolution satellite imagery from the Copernicus Sentinel-2 satellite. The primary objective is to develop a robust neural network model capable of accurately identifying water and land areas, thereby facilitating the detection and analysis of shoreline changes over time.

The model employed in this study is a Convolutional Neural Network (CNN) built on the U-Net architecture, originally proposed by (Ronneberger, Fischer, & Brox, 2015). The U-Net model is chosen for its effectiveness in semantic segmentation tasks, particularly in distinguishing between water and land in satellite images. By leveraging deep learning techniques, the model is trained on a specialized dataset comprising satellite imagery encompassing coastal regions, terrestrial landscapes, and water bodies across Europe.

The selected dataset is designed to enhance the U-Net model's ability to generalize across diverse coastal environments, ensuring accurate and consistent segmentation results. This capability is crucial for monitoring coastal dynamics and quantifying changes in coastline geometry over time. The neural network outputs are utilized to calculate variations in coastline measurements, providing valuable insights into coastal erosion (land loss), coastal accretion (land gain) and environmental impacts.

By integrating advanced machine learning with satellite remote sensing, this research aims to contribute to the scientific understanding and effective management of coastal zones. The use of the U-Net model highlights its potential to enhance strategies aimed at detecting shorelines between land and water bodies. This thesis highlights the importance of using advanced machine learning technology to monitor coastal areas, thereby promoting sustainable practices in coastal management amid global environmental challenges.

Shorelines and Shoreline Changes

Understanding Shorelines

The shoreline or Coastline is an imaginary line that marks the border between land and sea. The dynamic region, where land meets the sea, continuously changes due to natural forces like waves, tides, storms, and human activities. These forces have driven shoreline fluctuations, including erosion and accretion, for centuries and will continue to do so forever (Charles & O'Neill, 1985).

Shoreline Fluctuations

Shoreline changes or shoreline fluctuations can happen over various time scales, from long-term periods spanning years, decades, and centuries, to seasonal variations that occur annually, and short-term changes over days and months. These changes also vary in spatial scale, from meters to many kilometers (Pianca, Holman, & Siegle, 2015). The morphological evolution of coastlines accelerates significantly during extreme events like tsunamis, storms, and tropical cyclones, which can result in irreversible changes. Human activities also have a substantial impact, through the deliberate exploitation of coastal resources and as an unintended consequence of actions that degrade the coastal environment (Mentaschi, Vousdoukas, Pekel, Voukouvalas, & Feyen, 2018).

As noted by Salghuna and Bharathvaj (Salghuna & Bharathvaj, 2015), shoreline changes can generally be classified into two types based on their fluctuations: Erosion, Accretion, and Equilibrium where the shoreline extends seaward. Coastal erosion involves the removal of material from the shoreline, resulting in land loss as the shoreline moves inland. In contrast, accretion refers to the accretion of material at the shoreline, leading to land gain as the coast extends seaward. These states are influenced by five main factors: the amount of sediment reaching the beach, the energy of the waves, changes in sea level, the position of the shoreline, and human alterations to the coastline. Last, Equilibrium is when the shoreline remains stable.

The extraction of coastlines and water bodies is essential for fields like coastline change detection, coastal zone management, watershed delineation, and flood forecasting. This process is challenging and time-intensive, and it may be impractical for large regions such as entire coastal systems using traditional ground survey methods. Remote sensing data provide valuable preliminary estimates of changes and are indispensable for researching and monitoring coastal environments. Furthermore, as Kuleli Stated in 2010 in the Journal “Environmental Monitoring and Assessment”, due to the inaccuracy and outdated nature of existing maps, tracking rapid changes in coastlines can be difficult (Kuleli, 2010).

Remote Sensing

Introduction to Remote Sensing

As explained in Eric D. Conway's “An Introduction to Satellite Image Interpretation” (Conway, 1997), remote sensing refers to the practice of studying objects without direct physical contact, instead making measurements of their physical properties from a distance.

Satellite technology exemplifies remote sensing, as satellite sensors are specifically designed to analyze the energy reflected or emitted from the Earth's surface. By receiving data transmitted from satellites orbiting the Earth, individuals at ground-based receiving stations can gather detailed information about Earth's characteristics and environments without the need for direct on-site measurements. The fundamental principle of remote sensing is electromagnetic radiation. Electromagnetic radiation is the energy emitted in waveforms by substances that are above absolute zero temperature (-273,15°C or -459,67°F). Unlike material objects, a wave of radiation has no mass but can transmit energy from one location to another.

The Electromagnetic Spectrum

As stated in the NASA Science article on the electromagnetic spectrum (National Aeronautics and Space Administration, 2024). The electromagnetic spectrum encompasses a wide range of electromagnetic radiation types. This spectrum categorizes energy based on wavelength and frequency:

- **γ-rays** (wavelengths less than 0.01 nm and frequencies greater than 30 EHz), are used in cancer treatment, the sterilization of medical equipment, and astronomical
- **X-rays** (wavelengths from 0.01 nm to 10 nm and frequencies between 30 PHz and 30 EHz), are essential in medical imaging, security scanning, and crystallography
- **Ultraviolet radiation (UV)** (wavelengths of 10 nm to 380 nm and frequencies from 770 THz to 30 PHz)
- **Visible light** (wavelengths from 380 nm to 700 nm in and frequencies 430 THz to 770 THz) includes colors from red to violet and is crucial for human vision, photography, and illumination., is employed in sterilization, fluorescent lights, and the detection of forged banknotes.
- **Infrared radiation** (wavelengths from 700 nanometers to 1 millimeter and frequencies from 300 GHz to 430 THz) serving applications such as remote controls, thermal imaging, and night-vision equipment
- **Microwaves** (wavelengths from 1 millimeter to 1 meter and frequencies between 300 MHz and 300 GHz) are utilized in microwave ovens, radar, and communication devices
- **Radio waves** (with wavelengths greater than 1 meter and frequencies less than 300 MHz) are used in radio and television broadcasting, cell phones, and wireless networks.

Atmospheric Interaction

The atmosphere blocks short-wavelength and high-energy radiation like γ-rays, x-rays, and ultraviolet because of the atmospheric ozone. Ozone in the atmosphere absorbs nearly all radiation with wavelengths less than about 0.3 μm (0.003 mm). Without the ozone layer, much of this radiation would reach the Earth's surface, posing severe risks to living organisms and potentially making survival difficult. The atmosphere also absorbs energy within wavelength ranges, notably in the infrared and microwave segments of the electromagnetic spectrum. These specific wavelengths, absorbed by atmospheric gases, are referred to as absorption bands (Conway, 1997).

Application in Remote Sensing

In addition to absorption bands, there are regions in the electromagnetic spectrum where the atmosphere allows specific wavelengths to pass through unhindered. These wavelength ranges are called atmospheric windows, as they facilitate the transmission of radiation through the atmosphere. A significant atmospheric window exists in the visible light spectrum, coinciding with the peak of solar energy. On clear days, without clouds, most of the sun's visible light can pass through the atmosphere without being absorbed by atmospheric gases. Furthermore, there are multiple atmospheric windows in the infrared spectrum. These windows allow infrared radiation at specific wavelengths to penetrate the atmosphere. Particularly crucial for remote sensing, these infrared windows align with the peak wavelengths of thermal radiation emitted by the Earth. They allow thermal energy to pass through the atmosphere and reach space, where satellite sensors can capture and analyze it (Conway, 1997).

Types of Earth Observation Sensors

According to Chuvieco, earth observation sensors come in several types: photographic sensors, scanning imaging sensors, radar imaging sensors, and non-imaging sensors. Photographic sensors operate like digital cameras. Scanning imaging sensors capture two-dimensional images by scanning each point and line in sequence over time, they are widely used today and can be further divided into surface scanning and image scanning sensors. Radar imaging emits electromagnetic waves to create a lateral profile (Chuvieco, 2020). Currently, most Earth observation satellites are one of two primary categories: Synthetic Aperture Radar (SAR) satellites and optical satellites (photographic, scanning) that cover specific spectral bands.

Spectral Bands

As stated in “Chapter 3: Environmental applications of remote sensing” in the Book “Pollution Assessment for Sustainable Practices in Applied Sciences and Engineering” from (Mertikas, Partsinevelos, Mavrocordatos, & Maximenko, 2021), the visible and near-infrared regions of the electromagnetic spectrum are widely utilized as key spectral bands in remote sensing because they correspond to the highest levels of solar radiation, making them more prevalent than other spectral regions. Early advancements in sensor technology were primarily based on these bands, reinforcing their significance. In remote sensing, sensors detect energy that is either emitted or reflected by a target and record various signal properties, including intensity and characteristics across different spectral bands. By examining these radiometric and spectral properties, the chemical and physical attributes of the target can be determined. The reflectivity of a surface target is influenced by its chemical composition and crystal structure through molecular and electrical interactions with incoming radiation. Moreover, surface features such as roughness, slope, orientation, and the geometry of the source-surface-detector configuration also impact the reflected radiation. Consequently, analyzing the properties of the reflected electromagnetic signal captured by sensors provides detailed insights into the target's composition, geometry, and other characteristics.

Satellite Technology and Applications

Introduction to Satellites

Satellites are human-made objects placed in orbit around the Earth or other celestial bodies that can continuously monitor the entire globe or a designated portion of it over a specified time frame. Satellites are outfitted with an array of instruments and sensors designed to gather and relay data back to Earth. In 1957 was the launch of Sputnik 1, the first artificial satellite (Dickson, 2001). Since then, satellites have become crucial assets in communication, weather forecasting, navigation, and scientific research. Modern satellites are equipped to capture and transmit satellite images that monitor an extensive range of variables, including atmospheric gases, sea surface temperatures, urban development, and forest cover (Roddy, Jones, & Long, 2006).

Satellite Sensors and Types

According to Sommervold, Gazzea, & Arghandeh satellite sensors are designed to monitor distinct segments of the electromagnetic spectrum and are broadly categorized into two types: active and passive. Passive sensors utilize reflected sunlight for illumination, whereas active sensors employ their own source of illumination (Sommervold, Gazzea, & Arghandeh, 2023).

Synthetic Aperture Radar (SAR) satellites have active sensors equipped with onboard microwave energy sources. They operate using long-wavelength microwave radiation, which remains unaffected by weather conditions, enabling continuous image acquisition. SAR technology excels in providing detailed structural information about surface objects. However, the backscatter received by SAR sensors often introduces speckle noise, which can reduce image clarity and detail. Moreover, SAR sensors only capture the intensity of backscattered signals, resulting in monochromatic imagery devoid of spectral information (Sommervold, Gazzea, & Arghandeh, 2023). This limitation significantly impacts SAR's utility in Earth observation applications that rely heavily on spectral data for comprehensive analysis and interpretation, despite its capability to deliver high-resolution surface detail and robustness.

In contrast, **Optical Satellites** have passive sensors equipped, which utilize solar illumination for imaging, capturing reflected sunlight from objects on Earth's surface. This dependency on sunlight makes optical imagery susceptible to weather conditions and varying lighting conditions throughout the day. Optical satellites operate across specific spectral bands of the electromagnetic spectrum, commonly including red, green, blue, and near-infrared (NIR) wavelengths. Data from these bands can be combined to create various types of imagery and specialized indices for detailed terrain analysis (Sommervold, Gazzea, & Arghandeh, 2023). Applications such as vegetation maritime and infrastructure monitoring, and forest assessment, benefit significantly from these spectral capabilities, making optical satellite imagery the preferred choice in many remote sensing applications.

Satellite Programs and Agencies

According to the World Meteorological Organization’s OSCAR (Observing Systems Capability Analysis and Review) Tool (World Meteorological Organization, 2024), there are currently **357 operational satellites** (World Meteorological Organization, OSCAR - Satellites, 2024), launched by **124 satellite programs** (World Meteorological Organization, OSCAR - Satelite Programmes, 2024) and managed by **104 different agencies** (World Meteorological Organization, OSCAR - Space Agencies, 2024). One of them is the European Space Agency.

European Space Agency and Copernicus

The European Space Agency (ESA) is an intergovernmental organization dedicated to space exploration and research. Founded in 1975, the European Space Agency collaborates with 22 member states to promote Europe's capabilities in space technology and science (European Space Agency, About ESA, 2024). With its headquarters in Paris, France, ESA also operates several major facilities across Europe, including the European Space Operations Centre (ESOC) in Darmstadt, Germany; the European Space Research and Technology Centre (ESTEC) in Noordwijk, Netherlands; and the European Space Astronomy Centre (ESAC) in Villanueva de la Cañada, Spain. ESA carries out a variety of missions including Earth observation, human spaceflight, planetary exploration, and satellite navigation (European Space Agency, ESA Missions, 2024).

One of ESA's core programs is **Copernicus**, developed in collaboration with the with the Member States, the European Space Agency (ESA), the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT), the European Centre for Medium-Range Weather Forecasts (ECMWF), EU Agencies and Mercator Ocean International. Copernicus is a major Earth observation initiative, which provides essential data for environmental management, climate change monitoring, and civil security (Copernicus, 2024). This program relies on a fleet of Sentinel satellites illustrated in Figure 1 by ESA (European Space Agency, Copernicus Programm, 2024) and an extensive ground infrastructure to provide accurate and accessible information. These efforts demonstrate the ESA's commitment to advancing scientific understanding and supporting societal needs through innovative space-based technologies and applications.

Figure 1: Earth observation missions developed by ESA. [Credits: ESA]

Sentinel Mission

In order fulfill the requirements of the Copernicus program, the European Space Agency (ESA) has launched several innovative missions under the Sentinel program. These missions, which consist of satellite constellations, are positioned to offer frequent visits and extensive coverage, which are necessary to gather reliable data for Copernicus services. Modern technologies, such as radar and multispectral imaging equipment, are integrated into every Sentinel mission and are crucial for monitoring the Earth's landmasses, oceans, and atmosphere (European Space Agency, Copernicus Programm, 2024). Current active Copernicus satellites are Sentinel-1, Sentinel-2, Sentinel-3, Sentinel-5P.

Sentinel-1 is vital to this constellation, which consists of two sun-synchronous polar-orbiting satellites. These satellites share the same orbital plane but are phased 180° apart. Functioning continuously, they use C-band SAR imaging to capture images in any weather condition. The most recent addition, Sentinel-1B, was launched in April 2024, with Sentinel-1C scheduled for launch in December 2024 (European Space Agency, Sentinel-1, 2024).

Sentinel-2 specializes in high-resolution imaging of Earth's land surfaces, using a Multi-Spectral Instrument that captures 13 spectral bands: four at 10 m, six at 20 m, and three at 60 m spatial resolution. This allows for detailed observations of vegetation, soil, water bodies, and coastal areas, supporting applications such as emergency management and environmental assessments. The mission consists of two twin satellites, Sentinel-2A and Sentinel-2B, which move in the same orbit with a phase difference of 180°, achieving a revisit frequency of 5 days at the Equator. Sentinel-2A was launched in 2015, followed by Sentinel-2B in 2017. In September 2024, Sentinel-2C was also launched, expanding the constellation's capabilities (European Space Agency, Sentinel-2, 2024).

Sentinel-3 expands its mission scope by utilizing instruments designed to measure sea-surface topography, land and ocean surface temperatures, and the color of Earth's land and oceans. These include the Sea and Land Surface Temperature Radiometer (SLSTR), the Ocean and Land Color Instrument (OLCI), and the Altimetry Surface Topography Mission (STM) payload. These measurements are essential for predicting ocean behavior and monitoring environmental changes linked to climate. Sentinel-3A was launched on February 16, 2016, followed by Sentinel-3B on April 25, 2018, (European Space Agency, Sentinel-3, 2024).

Sentinel-5P, also known as Sentinel-5 Precursor, is dedicated to monitoring trace amounts of gases and aerosols that affect air quality and climate. Launched in 2017, Sentinel-5P carries the TROPOspheric Monitoring Instrument (TROPOMI), which was developed with funding from ESA and the Netherlands (European Space Agency, Sentinel-5P, 2024).

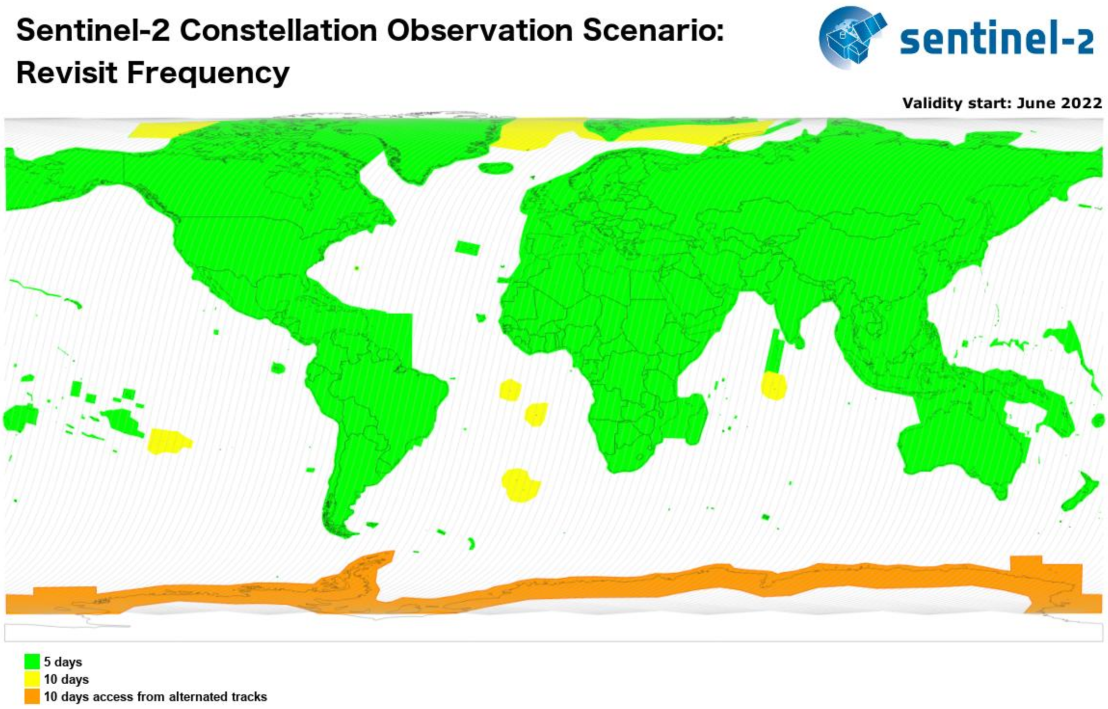
Sentinel-2 Mission

For this thesis, we will utilize data from the Optical Satellites of the Sentinel-2 mission. We selected Sentinel-2 because, according to the European Space Agency, its data is particularly well-suited for land and maritime monitoring applications (European Space Agency, Sentinel-2 Applications, 2024).

Geographical Coverage of Sentinel-2

As we can see in *Figure 2* by ESA (European Space Agency, Sentinel-2, 2024) the Sentinel-2 mission provides extensive geographic coverage, monitoring all continental land surfaces from the South Pole to 82.8° North latitude, including inland waters. It also covers coastal waters up to 20 kilometers from the shore, islands larger than 100 square kilometers, and all EU islands. Additionally, the mission includes the Mediterranean Sea and all enclosed seas like the Caspian Sea. *Figure 2* also illustrates the corresponding expected revisit times for Sentinel-2 acquisitions.

Figure 2: The coverage and revisit time foreseen for Sentinel-2 MSI acquisitions. [Credits: ESA]



Resolutions of the Sentinel-2

The Sentinel-2 mission, equipped with the Multi-Spectral Instrument (MSI), offers a variety of high-resolution data essential for numerous Earth observation purposes (European Space Agency, Senitnel-2 Scquisition Resolution, 2024). According to ESA, we define the main resolutions of the Sentinel-2 mission and its MSI instrument as follows:

The **temporal resolution**, or revisit frequency. This describes how often a satellite revisits a specific area. Each Sentinel-2 satellite revisits every 10 days, while the combined coverage of Sentinel-2A and Sentinel-2B reduces this frequency to 5 days.

The **spatial resolution** refers to the level of detail in ground representation captured by the satellite's sensors.

The **spectral resolution** measures the instrument's ability to distinguish different features in the electromagnetic spectrum, providing detailed spectral data for various applications.

The **radiometric resolution** determines the instrument's capability to distinguish incremental levels of intensity or reflectance, critical for detecting and analyzing fine details in intensity or reflectance. Increasing the radiometric resolution improves the system's ability to identify small variations in intensity or reflectance.

Spectral Bands available on Sentinel-2

Each Sentinel-2 satellite is equipped with a single Multi-Spectral Instrument (MSI), which includes 13 spectral channels covering the visible/near-infrared (VNIR) and short-wave infrared (SWIR) ranges (European Space Agency, Sentinel-2, 2024). *Table 1* from (Copernicus Sentinel-2 (processed by ESA), 2021) presents the spectral bands offered by the Sentinel-2 MSI sensor, detailing the central wavelength, bandwidth, and spatial resolution for each band.

Central wavelength is the midpoint of the range of wavelengths within a spectral band, measured in nanometers (nm), indicating the specific wavelength around which the band is centered. Bandwidth refers to the span of wavelengths that a spectral band encompasses, also measured in nanometers (nm). A narrower bandwidth suggests a more selective band that captures a smaller range of wavelengths, whereas a wider bandwidth covers a broader range.

Spatial resolution denotes the smallest size of an object or detail that a sensor can discern on the ground, typically measured in meters (m). Higher spatial resolution (represented by a smaller number, e.g., 10 meters) means the sensor can detect finer details.

Sentinel-2 bands	Sentinel-2A		Sentinel-2B		
	Central wavelength	Bandwidth (nm)	Central wavelength	Bandwidth (nm)	Spatial resolution (m)
Band 1 – Coastal aerosol	442.7	21	442.2	21	60
Band 2 – Blue	492.4	66	492.1	66	10
Band 3 – Green	559.8	36	559.0	36	10
Band 4 – Red	664.6	31	664.9	31	10
Band 6 – Vegetation red edge	704.1	15	703.8	16	20
Band 6 – Vegetation red edge	740.5	15	739.1	15	20
Band 7 – Vegetation red edge	782.8	20	779.7	20	20
Band 8 – NIR	832.8	106	832.9	106	10
Band 8A – Narrow NIR	864.7	21	864.0	22	20
Band 9 – Water vapour	945.1	20	943.2	21	60
Band 10 – SWIR – Cirrus	1373.5	31	1376.9	30	60
Band 11 – SWIR	1613.7	91	1610.4	94	20
Band 12 – SWIR	2202.4	175	2185.7	185	20

Table 1: Spectral bands of the Sentinel-2 MSI sensors

Introduction to Machine Learning

Machine learning is a computational process that uses input data to perform a specific task without being explicitly programmed to generate a predetermined result. These algorithms are inherently flexible, adjusting their structure iteratively through experience, thereby improving their ability to accomplish the task at hand. This adaptive process is known as training, where the algorithm learns from input data samples along with their corresponding desired outputs. Through this training, the algorithm optimizes its configuration to not only reproduce expected outcomes from the training data but also generalize to predict outcomes from new, previously unseen data. Importantly, this learning process is not confined to an initial phase but can continue to adapt and refine over time.

According to El Naqa and Murphy in the book “Machine Learning in Radiation Oncology: Theory and Applications” (El Naqa & Murphy, 2015) a successful Machine Learning algorithm offers two key benefits. Firstly, it can replace tedious and repetitive human tasks. Secondly, and more importantly, it has the potential to identify complex and nuanced patterns in the input data that may surpass the capabilities of an average human observer.

Supervised Learning

As stated in “Artificial Neural Network: A Brief Overview” by Zakaria, Mabrouka and Sarhan (Zakaria, Mabrouka, & Sarhan, 2014), supervised learning is a machine learning method that adjusts the parameters of an artificial neural network through labeled training data. The purpose is to determine the values of these parameters so that the network can produce correct outputs for any given input after training in the labeled data. The training data includes pairs of inputs and corresponding desired outputs, typically represented as data vectors. This method is also known as classification, and there are many classifiers available, each with its own advantages and disadvantages. Choosing the appropriate classifier (such as a multilayer perceptron, support vector machine, k-nearest neighbors’ algorithm, Gaussian mixture model, Gaussian, naive Bayes, decision tree, or radial basis function classifiers) for a specific problem is more a matter of art than science.

To solve a supervised learning problem, several steps must be followed. First, determine the type of training data required. Second, collect a training dataset that adequately represents the problem. Third, format the training data in a way that the chosen artificial neural network can understand. Fourth, train the network using this data. Finally, evaluate the performance of the trained network using a test (validation) dataset, which includes data not used during the training process.

Unsupervised Learning

Unsupervised learning is a machine learning technique that adjusts the parameters of an artificial neural network using data and a loss function that needs to be minimized. This loss function can be of any form and is determined by the specific task. Unsupervised learning is primarily used in applications of estimation problems such as statistical modeling, compression, filtering, blind source separation, and clustering. The purpose of unsupervised learning is to understand the structure of the data. Unlike supervised, the neural network in unsupervised learning receives only unlabeled examples. A common form of unsupervised learning is clustering, which categorizes data into groups based on their similarity. Among the various models of artificial neural networks, self-organizing maps often use unsupervised learning algorithms (Zakaria, Mabrouka, & Sarhan, 2014).

Semi-Supervised Learning

Semi-supervised learning combines elements from both supervised and unsupervised learning. In a semi-supervised dataset, the majority of data points are unclassified, but there are also a few data points that have been labelled. Semi-supervised models offer two main advantages: they achieve higher accuracy compared to unsupervised models when a few labelled data points are added, and they require less effort and time compared to fully supervised learning.

According to “Deep Learning Architectures” by Hosseini, Lu, Kamaraj, Slowikowski, and Venkatesh, semi-supervised learning is divided into transductive and inductive learning. In transductive learning, a classifier is first trained with labelled data. Afterwards, this classifier is employed to classify the complete set of unlabelled data. The most confident predictions for the unlabeled data, including their predicted labels, are usually incorporated into the training set. The classifier is then re-trained using this augmented dataset, and this iterative process, known as self-teaching or bootstrapping, continues until the model is ready to classify new test data points. Conversely, inductive learning focuses on deducing the relationship between input and output, aiming to generalize from the training data to predict outputs for new inputs (Hosseini, Lu, Kamaraj, Slowikowski, & Venkatesh, 2020).

Reinforcement Learning

Reinforcement learning, as defined by Sutton and Barto (Sutton & Barto, 1999), integrates principles from psychology’s trial-and-error tradition, engineering’s optimal control theory, and learning theories emphasizing secondary reinforcement and stimulus traces. This synthesis has spurred the development of computer algorithms within artificial intelligence aimed at learning policies that maximize long-term rewards for agents performing tasks. Originating in psychological concepts, these ideas have been extensively advanced in the artificial intelligence community. While not focused on experimental models in psychology or neuroscience, reinforcement learning remains pivotal in understanding learned, goal-directed behaviors, offering effective models for guiding actions based on incentives and utility.

Backpropagation

Backpropagation, introduced by Rumelhart, Hinton, and Williams in 1986 (Rumelhart, 1986), is a fundamental algorithm used to train artificial neural networks. At its core, backpropagation is an optimization technique that adjusts the weights of a network to minimize the difference between the predicted outputs and the actual targets. This process involves two main steps: the forward pass and the backward pass. During the forward pass, input data is passed through the network to generate predictions. In the backward pass, the algorithm calculates the gradient of the loss function (a measure of prediction error) for each weight by applying the chain rule of calculus. These gradients indicate how much each weight contributed to the error. The weights are then updated in the direction that reduces the error, typically using optimization techniques like gradient descent. By iteratively applying backpropagation, the network learns to improve its predictions and generalize from the training data. This approach enables the network to uncover complex patterns and relationships within the data, making it a cornerstone of modern machine learning and deep learning practices.

Deep Learning and Artificial Neural Networks

Deep Learning

Deep learning is a subfield of machine learning focused on algorithms inspired by the structure and function of the brain called artificial neural networks. It has gained immense popularity and attention in recent years due to its exceptional success in various domains such as image recognition, natural language processing, speech recognition, and autonomous driving.

Deep learning models are built using layers of artificial neurons, similar to the human brain's (biological) neural network, enabling them to learn and make intelligent decisions on their own. These models consist of multiple layers, hence the term "deep". Each layer in a deep learning model processes hierarchically input data, learning increasingly abstract representations. This hierarchical learning allows deep learning models to capture complicated patterns and relationships within the data.

Biological Neural Networks

The fundamental unit of a biological neural network is known as a neuron or nerve cell, that uses biochemical reactions to receive, process, and transmit information. The human brain contains more than 10 billion neurons in total.

As stated in “Neuroanatomy, Neurons” by Ludwig, Reddy and Varacallo (Ludwig, Reddy, & Varacallo, 2024), a neuron includes a cell body, or soma, where the cell nucleus is located. Connected to the cell body are dendritic fibres called dendrites, which receive signals from other neurons. Extending from the cell body is a long fibre known as the axon, which branches into multiple strands and connects to many other neurons at synaptic junctions, or synapses. The points of reception for these synapses are found both on the dendrites and on the cell bodies of other neurons. The axon of a typical neuron connects to several thousand synapses associated with other neurons.

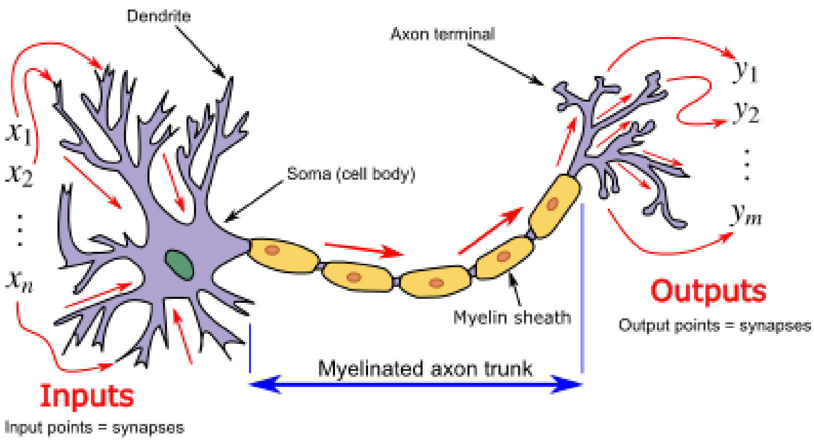


Figure 3: Illustration of a Biological Neuron by Prof. Loc Vu-Quoc

Figure 3 provides a precise illustration of a Biological Neural Network created by Prof. Loc Vu-Quoc (Vu-Quoc, 2019), as described in "Neuroanatomy, Neurons".

Artificial Neural Networks

Artificial Neural Networks (ANN) have likely been one of the most successful technologies of the past twenty years, widely utilized across a broad spectrum of applications in various fields.

An Artificial Neural Network is a mathematical model that attempts to simulate the structure and functions of biological neural networks. However, as stated by Yegnanarayana Bayya in his book "Artificial Neural Networks" (Yegnanarayana, 2009), these models are unlikely to achieve the same level of performance as their biological counterparts due to various significant challenges. Firstly, our understanding of how biological neurons function and how they are interconnected remains inadequate.

Additionally, it is extremely difficult to simulate the vast number of neurons, and their complex connections found in biological networks, as well as their ability to operate in natural asynchronous modes.

Artificial Neural Networks Methodology

Artificial Neural Networks or else simply Neural Networks are extensive parallel computing models consisting of many artificial neurons, as we can see in Figure 5. Weighted connections interconnect these numerous basic processors or artificial neurons, depicted in Figure 4. Analogously, these processing nodes can be likened to "neurons." Each node's output is determined exclusively by the local information available at that node, whether it is stored internally or received through its weighted connections. Each unit receives inputs from several other nodes and sends its output to additional nodes, as shown in *Figure 5*. Individually, each processing element has limited capability, producing a scalar output that is a straightforward non-linear function of its inputs (A.D.Dongare, R.R.Kharde, & D.Kachare, 2012).

Figure 4: A simple feedforward neural network with three layers and five inputs

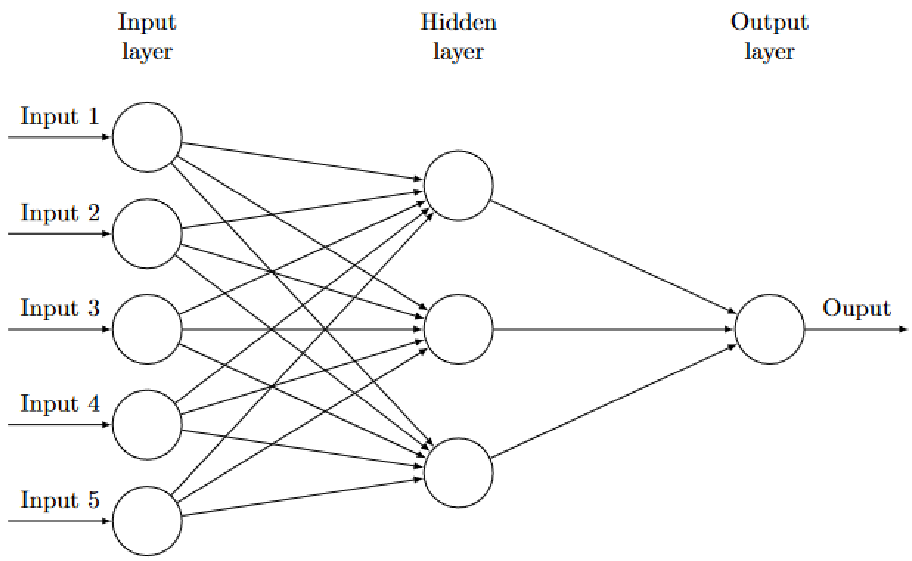
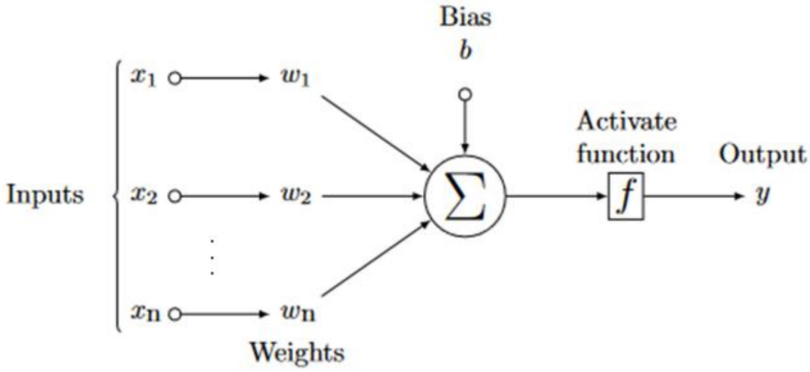


Figure 5: The internal workings of a single neuron in a neural network



Then, the neuron computes its impulse by summing the input signals (x_1, x_2, x_n) weighted according to their importance (w_1, w_2, w_n). It then applies a transformation using the transfer function, also known as the Activation Function. The artificial neuron's learning ability is facilitated by adjusting these weights based on the selected learning algorithm (Abraham, Artificial Neural Networks, 2005).

Artificial Neural Networks Layers

In a neural network, layers are critical elements that structure and direct the flow of information during the learning process. Each layer has a distinct role in processing and transforming data as it passes through the network.

The role of layers in a neural network is to transform input data through mathematical functions, defined by weights and biases, to produce meaningful outputs. During training, these weights and biases are adjusted based on the error between the predicted output and the actual target, allowing the network to learn patterns and relationships in the data. The structure and configuration of layers in a neural network are critical elements for the network's ability to generalize and operate efficiently on unseen data. As stated in “Artificial Neural Network Architectures and Training Processes” by (da Silva, Hernane Spatti, Andrade Flauzino, Liboni, & dos Reis Alves, 2017) , the fundamental structure includes three types of neuron layers:

The **Input Layer** is the first layer or features. Each node in this layer corresponds to a feature, and the number of nodes corresponds to the dimensionality of the input data. This layer receives information from the external environment. Inputs are usually normalized within the limits produced by activation functions, achieving better numerical precision for the network's mathematical operations.

Hidden Layers are the intermediary layers situated between the input layer and the output layer, performing most of the computations in a neural network. These layers contain neurons that extract patterns related to the process or system being analyzed. The number of hidden layers and the nodes in each layer are adjusted by hyperparameters, according to the complexity of the task.

The **Output Layer** generates the result or prediction of the neural network. The number of nodes in this layer varies according to the nature of the task. For binary classification, there may be one node, while for multi-class classification problems, there may be many nodes, one for each category.

There are also various types of layers in Artificial Neural Networks, each suited to different tasks and data structures:

Fully Connected Layers, where each node is connected to every node in the preceding and succeeding layers. This dense connectivity allows the network to learn intricate relationships within the data.

Convolutional Layers are designed for grid-like data, such as images. They perform convolutional operations to identify spatial hierarchies and extract features from the data. Convolutional layers are fundamental components of Convolutional Neural Network (CNN) architectures.

Pooling Layers, which are used in conjunction with convolutional layers. Pooling layers reduce the spatial dimensions of the data, which decreases computational complexity and highlights the most significant features.

Recurrent layers, known for being well-suited for sequential data, retain a hidden state that captures information from previous time steps. This allows the network to learn and leverage temporal dependencies.

Artificial Neural Networks Architecture

In **Feed-Forward Networks** (FNNs), data moves exclusively from input to output units in a single direction, without any feedback connections. Although data processing can occur across multiple layers, no connections originate from output units to input units within the same or earlier layers. FNNs are well-suited for tasks where data does not exhibit sequential patterns or temporal dependencies. Typical applications include tasks such as image classification (without spatial structures), handling tabular data, and solving regression problems.

Unlike feed-forward networks, **Recurrent Neural Networks** (RNNs) utilize feedback connections. These networks establish a direct relationship between network levels and specific positions in the sequence (timestamps). Instead of a fixed number of inputs to a single input layer, RNNs feature multiple layers, with each layer receiving input corresponding to a specific timestamp. RNNs are specialized for handling sequential data, which makes them suitable for tasks like predicting time series, modelling languages, recognizing speech, and translating languages. Variants of RNNs, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), provide different capabilities for handling sequential information (Aggarwal, 2018).

Convolutional Neural Networks (CNNs) are discriminative deep learning architectures that are particularly effective in processing two-dimensional data, such as images and videos. Inspired by the organization of the animal visual cortex, as studied by Hubel and Wiesel in the 1960s, CNNs leverage the concept of receptive fields, where cells detect light in small, overlapping regions of the visual field. CNNs use shared weights through convolutions rather than general matrix multiplication, significantly reducing computational complexity and enabling direct input of raw images, bypassing the feature extraction step of traditional algorithms. CNNs were influenced by time-delay neural networks (TDNN), which share weights temporally to reduce computation. The hierarchical layer training in CNNs, optimized by standard backpropagation algorithms, allows for effective handling of spatial relationships, further minimizing the number of parameters and enhancing performance. Additionally, CNNs require minimal pre-processing and benefit from GPU-accelerated computing, facilitating efficient training. CNNs have been successfully applied in various domains, including handwriting and face recognition, behaviour and speech recognition, recommender systems, image classification, and natural language processing (NLP) (Liu, et al., 2017).

Autoencoders employ unsupervised learning to discover a compact representation of data for dimensionality reduction, where the input matches the output. They consist of three main components: input, output, and a hidden layer that compresses and reconstructs the data. This compression-decompression process involves encoding and decoding steps. Autoencoders, a type of feed-forward neural network, include an additional bias for calculating reconstruction errors. Once trained, they function like typical neural networks for generating activations. Operating unsupervised, autoencoders use unlabeled data to learn weights solely from input data, forming a compressed representation of it. (Hosseini, Lu, Kamaraj, Slowikowski, & Venkatesh, 2020) There are various types of autoencoders that exist, such as vanilla autoencoders with three layers, multilayer autoencoders with additional hidden layers for increased complexity, and convolutional autoencoders for processing image data in 3D vectors.

Configuring a neural network involves setting it up so that applying a set of inputs yields the desired outputs. Methods for adjusting connection strengths involve setting weights based on prior knowledge or training the network by exposing it to patterns and modifying weights according to a specific learning rule. (Abraham, Nature and Scope of AI Techniques, 2005).

Activation Functions

As mentioned in Artificial Neural Networks

Artificial Neural Networks (ANN) have likely been one of the most successful technologies of the past twenty years, widely utilized across a broad spectrum of applications in various fields.

An Artificial Neural Network is a mathematical model that attempts to simulate the structure and functions of biological neural networks. However, as stated by Yegnanarayana Bayya in his book "Artificial Neural Networks" , these models are unlikely to achieve the same level of performance as their biological counterparts due to various significant challenges. Firstly, our understanding of how biological neurons function and how they are interconnected remains inadequate.

Additionally, it is extremely difficult to simulate the vast number of neurons, and their complex connections found in biological networks, as well as their ability to operate in natural asynchronous modes.

Artificial Neural Networks Methodology on *page 14* Activation Functions are responsible for transforming the weighted sum of inputs into an output signal that is crucial for learning and decision-making processes within the network. By applying these functions, neural networks can model intricate relationships and capture complex patterns in data, enhancing their capability to solve sophisticated tasks such as image recognition, natural language processing, and predictive analysis. Furthermore, the adjustment of weights through specific learning algorithms, facilitated by Activation Functions, enables continuous refinement and improvement of the network's predictive accuracy and performance.

As Hecht-Nielsen stated (Hecht-Nielsen, 1989), in 1958 Rosenblatt introduced the mechanics of the single artificial neuron. *Figure 4* is the schematic diagram of an artificial neuron, where $\{x1, x\mathbf{2}, \cdots, xn\}$ are the inputs of the artificial neuron, {w1, w2, ⋯, wn} are the weights corresponding to the inputs, b is the bias, and the addition unit ∑ gets the linear weight sum of the inputs $\{x\mathbf{1}, x\mathbf{2}, \cdots, xn\}$ and the bias. Denote $x = [x\mathbf{1}, x\mathbf{2}, \cdots, xn] \in Rn$ and $w = [w\mathbf{1}, w\mathbf{2}, \cdots, wn] \in Rn$, then the output ∑ of the addition unit can be represented as:

$$\sum = xw^T + b$$

The Function f referred to as the Activation Function is used to simulate the response state of a biological neuron and obtain the output y.

$$y = f(\sum)$$

In neural networks, various activation functions are employed, each with distinct characteristics suited for different tasks. Below are several commonly employed activation functions:

The **Sigmoid Function**:

$$\boldsymbol{\sigma(x) = \frac{1}{1 + e^{-x}}}$$

Sigmoid functions are commonly used in binary classification tasks where the output must be within the range of 0 to 1, representing probabilities. They are also used in the output layer of neural networks for binary classification problems. According to "Activation Functions and their Characteristics in Deep Neural Networks by Ding (Ding, 2018), the main advantage of the sigmoid function is its smooth, continuous output, making it suitable for gradient-based optimization methods like gradient descent. However, it suffers from vanishing gradient problems, which can slow down the training of deep neural networks.

According to Hochreiter in the Article "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions" (Hochreiter, 1998), the vanishing gradient problem is a phenomenon that occurs during the training of deep neural networks, especially networks with many layers. It refers to the situation where the gradients become extremely small as they propagate backwards through the network during the process of backpropagation.

The **Hyperbolic Tangent (Tanh):**

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Tanh functions are often used in hidden layers of neural networks where inputs are normalized within a range of -1 to 1. It is particularly useful in scenarios where the output needs to be zero-centred, which can aid in optimization in certain cases. However, like sigmoid, it also suffers from the vanishing gradient problem for very large or very small inputs.

The **Rectified Linear Unit (ReLU)**:

$$ReLU(x) = \max(0, x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

ReLU is widely used in hidden layers of deep neural networks due to its simplicity and effectiveness in training. It helps in mitigating the vanishing gradient problem and speeds up convergence during training. ReLU is computationally efficient and does not activate all neurons simultaneously, which helps in sparse activations, aiding the network's ability to learn different features. However, it can lead to dead neurons (neurons that never activate) during training, especially with large learning rates.

Loss Functions

Loss functions are crucial components in machine learning algorithms, especially in supervised learning tasks where models learn from labeled data. Their primary goal is to evaluate how closely the model's predictions align with the actual labels in the training dataset. Essentially, a loss function also referred to as a cost function, provides a single numerical value that indicates how effectively the model is performing based on the given parameters and weights.

Different types of loss functions are used depending on the task at hand. For regression tasks, such as predicting continuous values, popular choices include Mean Squared Error (MSE) and Mean Absolute Error (MAE), which quantify the average squared and absolute differences between predicted and actual values, respectively. In contrast, for classification tasks like distinguishing between two classes, Binary Cross-Entropy (Log Loss) is commonly employed to evaluate the model's performance by penalizing incorrect predictions based on their probabilities.

During the training process, the objective is to minimize the value of the loss function through optimization techniques like Gradient Descent. This iterative approach adjusts the model parameters to reduce errors and improve the overall fit of the model to the training data. The lower the loss value, the better the alignment between predicted outcomes and actual observations, indicating higher accuracy and reliability of the model.

Creating the Dataset from Sentinel-2 Images

As discussed on page 8 in Chapter: Sentinel-2 Mission, this thesis leverages data from the Sentinel-2 mission, chosen for its effectiveness in land and maritime monitoring, as the European Space Agency recommended.

Image Collection Process

For the Collection of the Data, we utilize the Sentinel-Hub API in Python programming language by Sinergise (Sinergise, 2017) through a CDSE (Copernicus Data Space Ecosystem) account (Copernicus, (ESA), 2024). Sentinel Hub is a “multi-spectral and multi-temporal big data satellite imagery service, capable of fully automated archiving, real-time processing and distribution of remote sensing data” (Sentinel-Hub, 2024). We use this API to retrieve satellite data over our Area of Interest (AOI) and specific time range.

As an example, we have randomly collected the dataset, which encompasses satellite imagery from the coastlines of Greece, Turkey, the Black Sea, and Italy, selected for their ecological significance and diverse coastal characteristics. The images are sourced between 2016 and 2023, corresponding to the available data from Sentinel-2. For each bounding box coordinate, an image is selected from this timeframe based on minimal cloud coverage. This approach ensures that the dataset includes imagery with optimal clarity and fidelity for analysis, capturing seasonal variations and long-term environmental changes relevant to coastal monitoring.

Our method for selecting bands and creating evalscripts (explained on *page 22*) aims to produce accurate and visually interpretable imagery tailored to our areas of interest and dataset labelling. For True Color RGB images, we used a predefined evalscript named “Sentinel-2-L2A True Colored Optimized” from Sentinel-Hub (Sentinel-Hub, custom-scripts.sentinel-hub, 2024). This script uses the B04 (Red), B03 (Green), and B02 (Blue) bands. This combination mimics human vision and enables detailed land features, vegetation, and coastal dynamics analysis. We preferred the “Optimized True Color” because it effectively enhances the overall visual diversity of the images. By implementing highlight compression, it ensures that bright areas, such as clouds or snow, do not max out, preserving details that might otherwise be lost. This is crucial for maintaining the nuanced information necessary for accurate image analysis and classification. Additionally, using sRGB (standard RGB) encoding addresses the issue of shadow regions becoming excessively dark, which can obscure important features. This encoding standard ensures that the color representation remains consistent and realistic across different devices and viewing conditions. The small saturation boost further enriches the color representation, making subtle variations more pronounced and thereby increasing the perceptual diversity of the images. This will prevent the dataset from the problem of low diversity or low variance.

By comparing the histograms of Optimized True Color and Simple True Color across HSV (Hue, Saturation, Value) and RGB (Red, Green, Blue) channels from the Dataset, we can observe the enhancements and optimizations made to the images. On Figure 6 we can see the histogram of the RGB Channels from the entire Dataset collected by using the “Optimized True Color Evalscript”. On the other hand, on Figure 7 we can see the histogram of the RGB Channels from the entire Dataset collected by using the “True Color Evalscript” (Sentinel-Hub, True Color, 2024).

Figure 6: Histogram of Optimized True Color RGB Channels

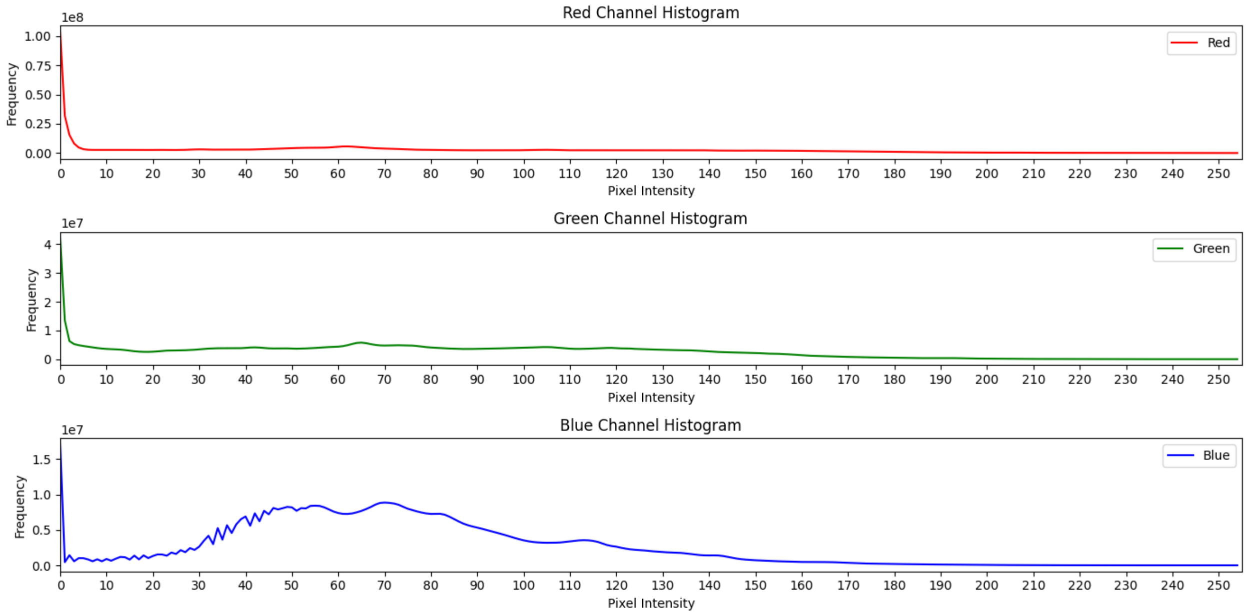
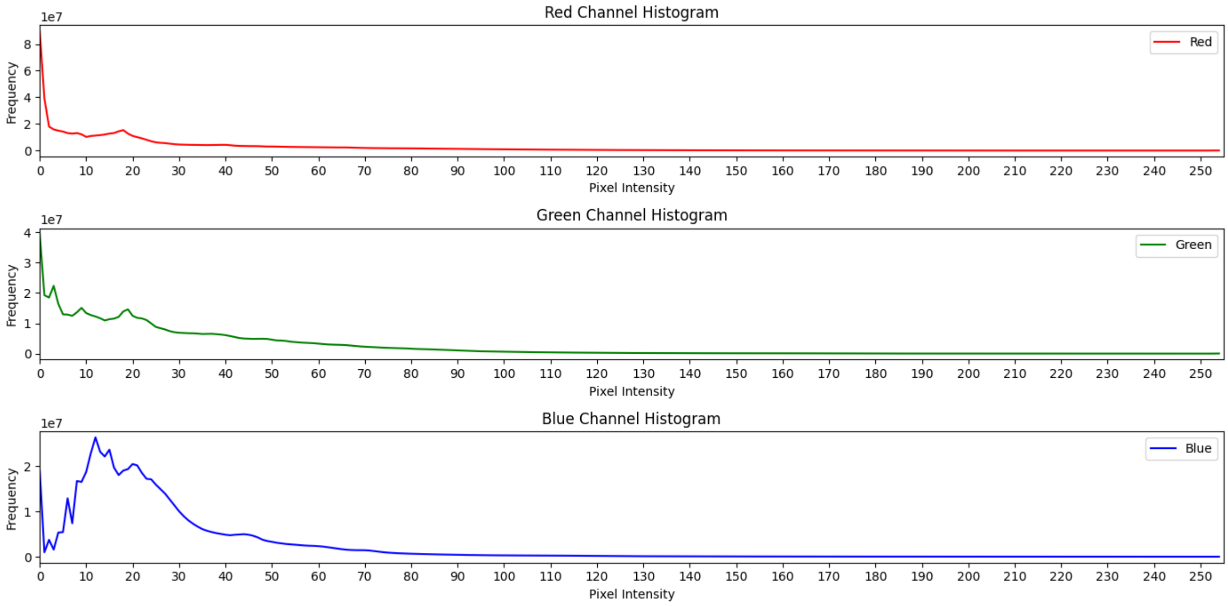


Figure 7: Histogram of Simple True Color RGB Channels



Similarly, in Figure 8, we present the histogram of the HSV channels for the entire dataset obtained using the "Optimized True Color Evalscript." Conversely *Figure 8* illustrates the histogram of the HSV channels for the entire dataset collected using the "True Color Evalscript".

Figure 8: Histogram of Optimized True Color HSV Channels

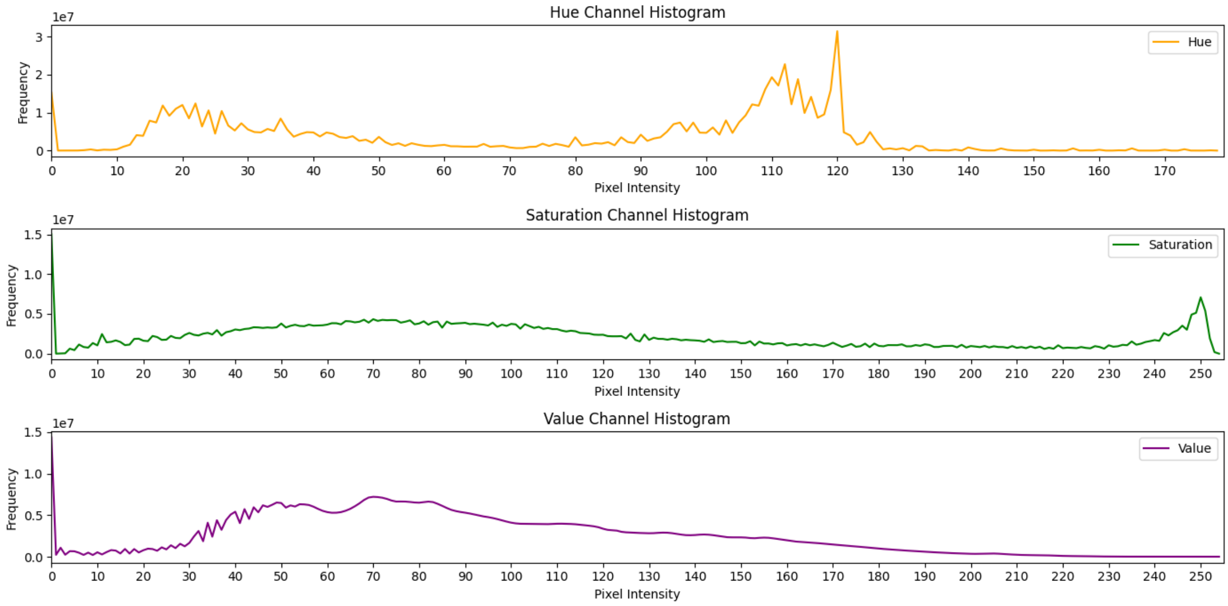
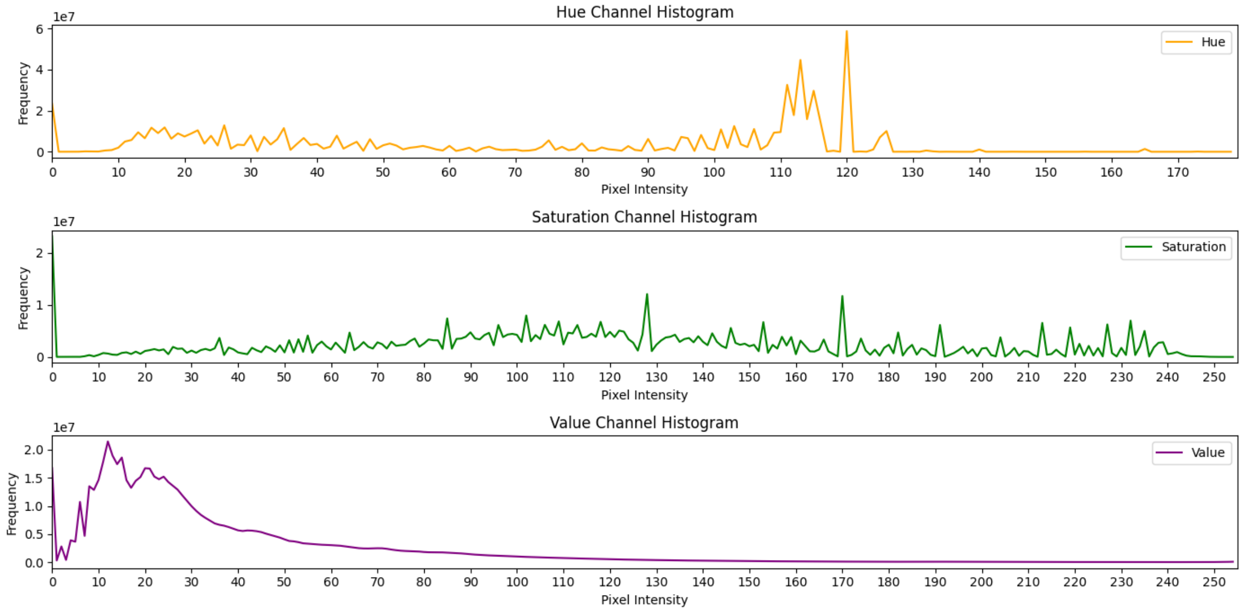


Figure 9: Histogram of Simple True Color HSV Channels



We can notice that the histograms of Optimized True Color show a more balanced distribution of colors and intensities. These differences highlight the effectiveness of the "Optimized True Color" approach in enhancing visual diversity and detail in the images, which can significantly improve the performance of image-based analyses and machine-learning models.

Additionally, for the Label Data, we customize the “Normalized Difference Water Index (NDWI)” evalscript (Sentinel-Hub, custom-scripts.sentinel-hub, 2024), which uses the B02 (Blue) and B08 (Near-Infrared) bands, ensuring grayscale imagery with white pixels representing land and black pixels indicating water bodies. To achieve binary segmentation, we also applied a threshold to the labelled data.

Through the Sentinel-Hub API, we also apply an up-sampling method to the retrieved satellite images, specifically nearest neighbor interpolation, selected for its ability to preserve pixel clarity essential for accurate differentiation of land and water features. Each of the 8000 images collected undergoes meticulous organization within structured data files and folders, facilitating efficient management and retrieval for subsequent analysis

What is an Evalscript?

According to Sentinel-Hub (Sentinel-Hub, Evalscript, 2024) an evalscript, or "custom script" is a JavaScript code segment that specifies how satellite data (bands) should be processed by Sentinel Hub and what values the service should return. It is an essential component for any process, batch processing, or OGC request. Evalscripts can utilize any JavaScript functions or language constructs, as well as various utility functions provided for ease of use.

Technical Specifications

Each image is captured at a resolution of 256x256 pixels, approximately 5 square meters per pixel, which is the smallest resolution by Sentinel-2 Satellites for detailed spatial analysis in coastal monitoring. To maintain the integrity of the squared image format required, geospatial boundaries are defined using the WGS84 Ellipsoid Earth Model. Based on this Earth Model we calculated the maximum latitude and longitude for a given minimum latitude and longitude, ensuring each captured image represents a precise 327,680 square meters area (256x256 pixels * 5 square meters per pixel).

What is the WGS84 Ellipsoid Earth Model?

The WGS84 (World Geodetic System 1984) **Ellipsoid Earth Model** (Wikipedia, Earth Ellipsoid, 2024), retrieved and maintained by the Office of Geomatics, National Geospatial-Intelligence Agency (Office of Geomatics, 2024), is a comprehensive mathematical representation of the Earth's shape and size, essential for global positioning and navigation. This model conceptualizes the Earth as an ellipsoid (a slightly flattened sphere to account) for the equatorial bulge caused by the planet's rotation. Key parameters of the WGS84 include a semi-major axis α (equatorial radius) of 6,378,137.0 meters and a semi-minor axis β (polar radius) of 6,356,752.314245 meters, with a flattening factor of approximately 1/298.257223563. The model also defines the first eccentricity squared:

$$e = \frac{1 - (\beta^2)}{\alpha^2}$$

Which is crucial for various geodetic calculations. Additionally, WGS84 encompasses both a reference ellipsoid and a geoid, the latter approximating mean sea level and serving as a more accurate reference for elevations. The WGS84 horizontal datum (a model used to precisely measure positions on Earth) provides a globally consistent coordinate system used extensively in GPS, GIS, aviation, and maritime navigation. This standardization ensures accurate mapping, navigation, and geospatial data integration worldwide, making WGS84 indispensable for modern geodetic and cartographic practices.

Calculating Coordinates

To account for the variation in the Earth's radius with latitude due to its ellipsoidal shape, we first calculated the Earth's radius at a given latitude (φ). This involved determining two radii of curvature: the Prime-Vertical Radius of Curvature (N(φ)) and the Meridian Radius of Curvature (M(φ)). The Prime-Vertical Radius of Curvature (Wikipedia, Earth Radius - Prime Vertical, 2024), which is the radius of curvature in the vertical plane relative to the meridian, is given by:

$$N(\varphi) = \frac{\alpha}{\sqrt{1 - e^2 \sin^2 \varphi}}$$

where α is the semi-major axis and e is the eccentricity of the ellipsoid.

The **Meridian Radius of Curvature** (Wikipedia, Earth Radius - Meridional, 2024), which is the radius of curvature in the plane containing the meridian, is calculated as:

$$M(\varphi) = \frac{1 - e^2}{\alpha^2} N(\varphi)^3$$

After calculating these radii, we used them to add the required distance to a given point, 1280 square meters in our case. Starting with the minimum longitude and latitude, we determined the maximum longitude and latitude to define the bounding box of the area of interest. This method allowed us to accurately account for the Earth's ellipsoidal shape in our geospatial calculations.

Training The Neural Network

Choosing the right architecture and training approach is crucial in machine learning projects, especially when dealing with complex data like satellite imagery. The selection of an appropriate neural network architecture can greatly affect the model's performance, efficiency, and ability to generalize from training data to unseen data.

Neural Network Selection

In this project, we chose to utilize a neural network, specifically a Convolutional Neural Network (CNN) and U-Net architecture, due to their demonstrated effectiveness in image processing tasks, particularly in the domain of semantic segmentation. CNN is well-known for its ability to automatically and adaptively learn spatial hierarchies of features from input images, making it an ideal choice for image classification and segmentation tasks. The U-Net architecture, designed by Ronneberger, Fischer and Brox for biomedical image segmentation (Ronneberger, Fischer, & Brox, 2015), enhances this capability by providing a symmetric expansion path which allows precise localization. This is particularly beneficial for satellite imagery especially where precise segmentation of land and water bodies is required.

U-Net Architecture

The U-Net architecture is composed of two primary components: a contracting path (encoder) and an expansive path (decoder). This design allows the network to capture both contextual information and detailed localization, making it well-suited for segmentation tasks that require precise classification of each pixel.

As illustrated on *Figure 10*, the path of the encoder (first 4 blocks) follows the standard architecture of a convolutional network. It involves repeated applications of convolutional layers and max-pooling layers. Specifically, two 3x3 convolutional layers are followed by a rectified linear unit (ReLU). These are then followed by a 2x2 max pooling operation with a stride of 2 for downsampling. At each downsampling step, the number of feature channels is doubled, allowing the network to capture more complex features as the spatial dimensions are reduced.

The path of the decoder aims to restore the spatial dimensions of the input image while preserving the learned features. Each step in the expansive path involves upsampling the feature map followed by a 2x2 convolution, which halves the number of feature channels. The upsampled feature map is then concatenated with the correspondingly cropped feature map from the contracting path, ensuring that high-resolution features are combined with the learned context. Following the concatenation, two 3x3 convolutional layers are applied, each followed by a ReLU. The final layer of the network is a 1x1 convolution that maps each 64-component feature vector to the desired number of classes.

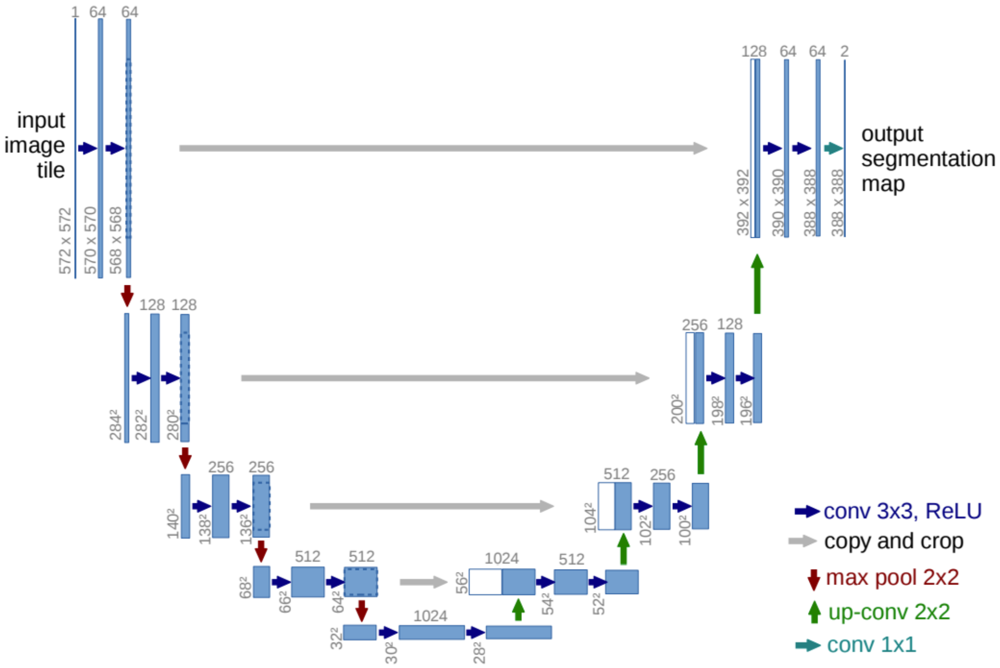


Figure 10: Each blue box represents a multi-channel feature map, with the number of channels indicated above the box. The x-y dimensions are shown at the lower left corner of each box. White boxes signify copied feature maps, and the arrows denote the various operations performed (Ronneberger, Fischer, & Brox, 2015)

Refined U-Net Architecture

The choice of the U-Net architecture is justified based on the nature of satellite imagery data, which involves identifying and segmenting objects in high-resolution images. The U-Net's ability to capture fine details and its effectiveness in segmenting images with fewer training samples make it a suitable choice for this project. After experimenting with the basic U-Net Architecture, we made some adjustments to enhance the model's accuracy when is fitted on our dataset. Specifically, we added more Layers on each Encoder and Decoder Block, the first one is a LeakyReLU layer, and the second one is a Batch Normalization layer.

LeakyReLU (Leaky Rectified Linear Unit) is a variation of the ReLU activation function. While ReLU converts all negative values to zero, LeakyReLU allows a small, non-zero gradient (controlled by a parameter, often set to 0.01) when the unit is inactive. This means that for negative input values, the LeakyReLU function will return a small negative slope instead of zero. This adjustment helps to mitigate the "dying ReLU" problem, where neurons can become inactive and produce only zeros, thus hindering the learning process. By adding LeakyReLU layers in each encoder and decoder block, we ensure that the network maintains some level of gradient flow even for negative inputs, enhancing the model's learning capacity and improving convergence during training (Maas, Hannun, & Ng, 2013).

Batch Normalization is a technique that normalizes the inputs of each layer to have a mean of zero and a variance of one. This normalization is applied to each mini batch, which aids in stabilizing and speeding up the training process. By reducing the internal covariate shift, Batch Normalization allows for higher learning rates, making the network less sensitive to initialization and reducing the need for dropout for regularization. Incorporating Batch Normalization layers in each encoder and decoder block helps to smooth the optimization landscape, leading to faster convergence and better generalization. It also provides a slight regularization effect, reducing the risk of overfitting by introducing some noise to each layer during training (Ioffe & Szegedy, 2015).

The refined U-Net architecture employed for our segmentation task is characterized by a significant number of parameters, which reflects its complexity and capacity to learn intricate patterns from the data. The total number of parameters in the model is 107,736,197, which corresponds to 410.98 MB of memory. Out of these, 35,910,785 are trainable parameters, occupying 136.99 MB. These trainable parameters include weights and biases that the model adjusts during the training process to minimize the loss function and improve performance. Additionally, the model has 3,840 non-trainable parameters, which take up 15.00 KB. These non-trainable parameters typically include fixed weights or biases associated with certain layers like Batch Normalization, which maintains statistics (mean and variance) that are not updated during training.

Furthermore, the optimizer parameters, which are crucial for the training process, amount to 71,821,572 and require 273.98 MB of memory. These parameters are used by the optimizer to adjust the trainable parameters iteratively based on the gradients computed during backpropagation. The substantial number of optimizer parameters highlights the importance of efficient optimization algorithms in handling such a large and complex model.

This extensive set of parameters enables the refined U-Net to capture detailed and diverse features from the satellite imagery, contributing to its high performance in segmentation tasks.

However, it also necessitates significant computational resources and memory, underscoring the need for powerful hardware, such as GPUs, to facilitate efficient training and inference.

Hardware and Software Setup

For training the neural network, we used an Ubuntu 24.04 LTS (Noble Numbat) Linux operating system, with a high-performance Graphics Card NVIDIA GeForce RTX 4090 (24GB of GDDR6X VRAM and 4^{th} Gen Tensor Cores), combined with an SSD 1TB PCIe Gen5 NVMe 2.0, to handle the computationally intensive IO tasks, and a RAM 64GB 6000MHz.

We also employed a combination of advanced software tools and frameworks to build, train, and ensure the reproducibility of our neural network model. We used **TensorFlow (2.16.1) Keras (3.3.3)**, an open-source machine learning framework in Python, that provides extensive libraries and tools that facilitate the construction of neural networks, efficient training using GPUs, and deployment in various environments. Its flexibility and scalability make it ideal for developing high-performance models for image segmentation tasks. (Tensorflow, 2024)

To maximize computational efficiency, we utilized **CUDA** (Compute Unified Device Architecture), a parallel computing platform and application programming interface model created by NVIDIA. CUDA enables TensorFlow to leverage the power of NVIDIA GPUs, significantly accelerating the training process by parallelizing computations. (NVIDIA, 2024)

Additionally, we used **Docker** to create a consistent and reproducible training environment. Docker allows us to encapsulate the entire training setup, including dependencies and configurations, into a container. This ensures that the training process can be replicated across different systems without discrepancies, thereby enhancing the reliability of our results. Docker's containerization also simplifies the deployment of the trained model to different production environments, ensuring that it runs seamlessly regardless of the underlying infrastructure. (Docker Inc., 2024)

Models Training Process

In the training process, we experimented with different versions of the U-Net network to identify the most effective architecture and settings for our satellite imagery segmentation task. The Preprocessed Satellite imagery data was split by 90% (7228 images) for the training section of the Network and 10% (803 images) for Testing purposes.

Simple U-Net with Binary Crossentropy

Initially, we implemented the original U-Net architecture as proposed by Ronneberger, Fischer, and Brox described in *page 24*. For this version, we used Binary Crossentropy as the loss function. Binary Crossentropy is a widely used loss function for binary classification tasks, which quantifies the difference between the predicted probability distribution and the actual distribution. The hyperparameters were set as follows: Epochs: 100, Learning Rate: 0.001, Batch Size: 16 (chosen to balance computational efficiency and model performance, as smaller batch sizes can often lead to better generalization, particularly in segmentation tasks), Steps per Epoch: 362. The results showed a very small Loss of 0.032 but an Accuracy of only 0.57. This discrepancy indicates that while the model was minimizing the loss function, it was not effectively capturing the details necessary for high accuracy in segmentation tasks.

To address the limitations of the Simple U-Net, we added LeakyReLU and Batch Normalization layers in each encoder and decoder block that were mentioned on page 25 at chapter "Refined U-Net Architecture". With these refinements, we used the same hyperparameters as before. The results showed a slight improvement in loss (0.028), but accuracy remained at 0.57.

Advanced Metrics for Segmentation

To better assess the performance of our model in the context of segmentation, we incorporated the additional metrics Dice Loss, Dice Coefficient, and Binary Intersection over Union which are more suitable for this specific task based on (Sudre, Li, Vercauteren, Ourselin, & Jorge Cardoso, 2017). These metrics provide a more comprehensive evaluation of how well the model segments the satellite imagery, considering both the spatial structure and the potential imbalance between classes.

Dice Loss is designed to measure the overlap between the predicted segmentation and the ground truth. It is calculated as:

$$Dice\ Loss = 1 - \frac{2 \times |A \cap B|}{|A| + |B|}$$

Where A is the predicted segmentation and B is the ground truth. Dice Loss penalizes both false positives and false negatives, making it especially valuable for imbalanced datasets where the background class may dominate. A lower Dice Loss value indicates better performance, with values close to 0 suggesting a high degree of overlap between the predicted and actual segmentations. For accurate results, Dice Loss values should ideally be below 0.1, indicating minimal discrepancy between the predictions and the ground truth.

Dice Coefficient is the complement of Dice Loss and is given by:

$$Dice\ Coefficient = \frac{2 \times |A \cap B|}{|A| + |B|}$$

It reflects the degree of similarity between the predicted segmentation and the ground truth, with values ranging from 0 to 1. A Dice Coefficient close to 1 indicates a high similarity and hence a good segmentation performance. For accurate results, the Dice Coefficient should be above 0.9, indicating a substantial overlap between the prediction and the ground truth.

Binary Intersection over Union (IoU) measures the overlap between the predicted and actual segmentations, defined as:

$$Binary\ IoU = \frac{|A \cap B|}{|A \cup B|}$$

It is a stricter metric than accuracy because it takes into account both the true positives and the false positives/negatives. IoU values range from 0 to 1, with values closer to 1 indicating better segmentation performance. For accurate results, IoU values should ideally be above 0.7, reflecting a substantial agreement between the predicted and actual segmentations.

While binary crossentropy is a common loss function for binary classification tasks, it is less effective for segmentation models. This is because binary crossentropy treats each pixel independently and does not account for the spatial structure of the segmentation map. This limitation can lead to misleadingly low Loss if the background class dominates, as it does not differentiate well between the importance of foreground and background regions. In contrast, Dice Loss, Dice Coefficient, and Binary IoU are tailored for segmentation tasks, providing a more balanced evaluation by focusing on the overlap and agreement between the predicted and actual segmentations. These metrics ensure that both foreground and background classes are adequately represented and penalize errors in the prediction of the segmentation map, leading to more reliable and meaningful performance assessments.

Overfitting and Hitting Plateau

Despite the improved metrics, the model was overfitting and hitting a plateau. Overfitting occurs when the model learns the training data too well, including its noise and outliers, leading to poor generalization to new data (Goodfellow, Bengio, & Courville, 2016). A training plateau happens when the model's performance stops improving despite continuing training, indicating that the learning rate might be too high or other optimization issues are present. (Schaul, Borsa, Modayil, & Pascanu, 2019)

Final Model with Additional Techniques

To address the issues of overfitting and plateauing performance, we implemented several advanced techniques to enhance our model. First, we applied data augmentation, introducing transformations such as rotation (up to 90 degrees), shear (up to 0.2), and horizontal flipping to increase the diversity of the training data and improve the model's generalization capabilities. Next, we incorporated early stopping by monitoring the validation loss and halting training when performance ceased to improve, thereby preventing overfitting. Additionally, we employed a learning rate reduction strategy, reducing the learning rate when performance plateaued, with a maximum learning rate of 0.0001 and a minimum of 0.000001. These adjustments collectively led to a significant improvement in the model's performance. The final model achieved a Dice Loss of 0.0062, a Dice Coefficient of 0.993 a Binary Crossentropy of 0.089, and a Binary IoU of 0.341. Due to the new metrics, we can now observe the Accuracy of the model depending on the Dice Coefficient Metric.

Table 2: Metric Results of Each Experimented Model

Model	Dice Loss	Dice Coefficient	Accuracy	Binary Crossentropy	Binary IoU
Simple U-Net	-	-	0.57	0.032	-
U-Net with LeakyReLU & Batch Normalization	-	-	0.57	0.028	-
Final Best Model	**0.0062**	**0.993**	**-**	**0.089**	**0.341**

Summarized Performance Analysis of the Final Refined U-Net Model: The training results showed a very small Dice Loss and a very high Dice Coefficient, indicating excellent segmentation performance. The model did not exhibit signs of overfitting as the test results were also accurate as we can notice on *Figure 11: Test Segmentation Results*.

Table 3: Hyperparameters and Metrics Used on Each Experimented Model

Model	Epochs	Learning Rate	Batch Size	Steps per Epoch	Loss Function	Additional Metrics
Simple U-Net	100	0.001	16	362	Binary Crossentropy	Accuracy
U-Net with LeakyReLU & Batch Normalization	100	0.001	16	362	Binary Crossentropy	Accuracy
Final Model	**100**	**0.0001 (reduced)**	**16**	**362**	**Dice Loss, Binary Crossentropy**	**Dice Coefficient, Binary IoU**

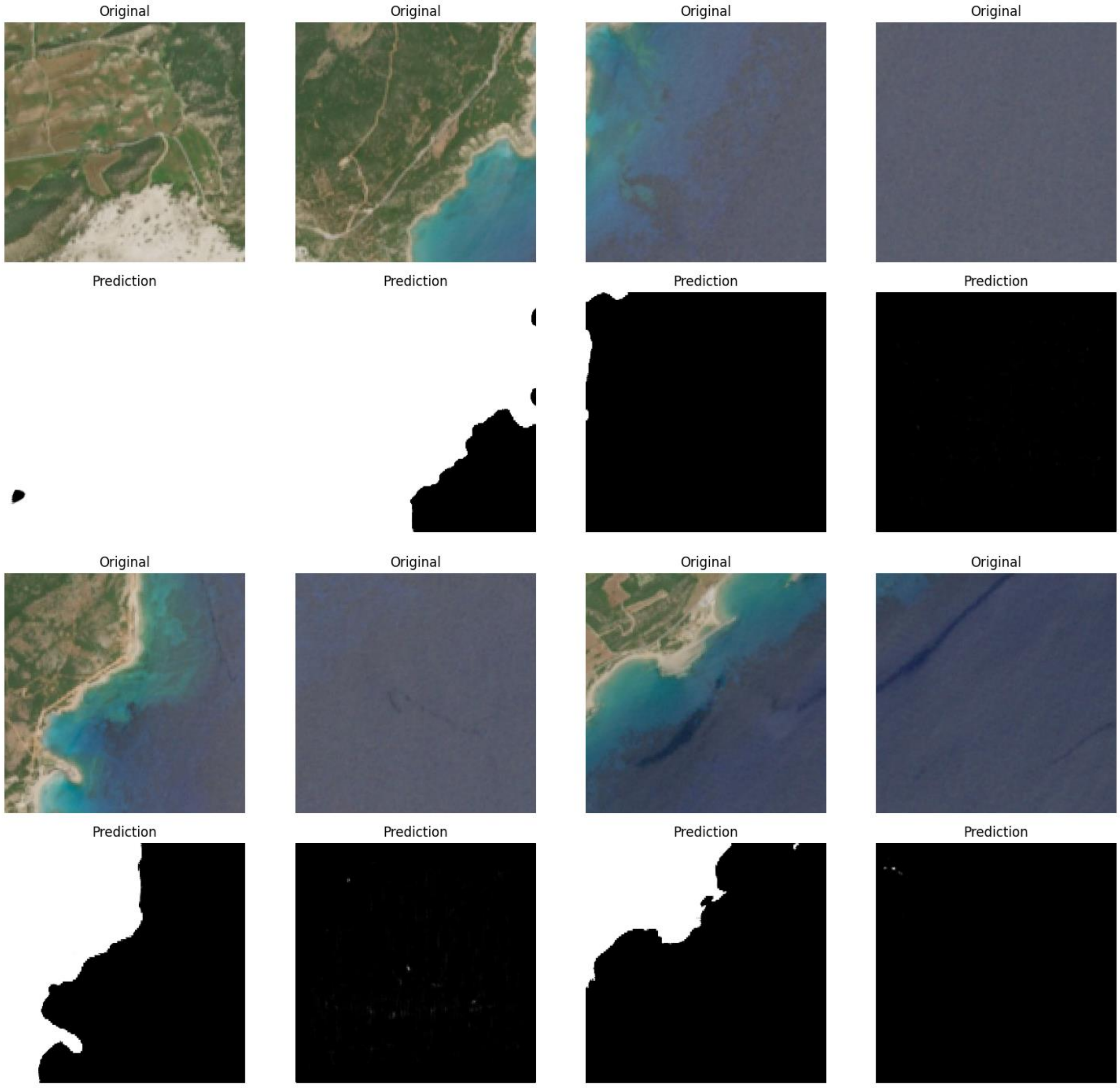


Figure 11: Test Segmentation Results

Main Flow: Using the Trained Model on Areas of Interest

After training the selected model, we developed a main workflow for applying the trained model to an Area of Interest (AOI). This workflow allows us to produce shorelines and calculate shoreline changes over a given year interval.

Collect Areas of Interest Satellite Images

The AOIs were collected from the Sentinel Hub API using a similar methodology as the dataset collection (page 19), based on specified minimum longitude and latitude coordinates. Each AOI covers a square area of 1024x1024 pixels, with a resolution of 5 square meters per pixel, resulting in a total area of 5,242,880 square meters. The maximum longitude and latitude coordinates were calculated using the Ellipsoid Earth Model, as detailed on *page* ***Error! Bookmark not defined.***.

Once the coordinates of the AOI were determined, we subdivided each AOI into 16 child AOIs. This subdivision was necessary to improve the accuracy of the model by providing it with smaller images, thus preventing resolution noise that could interfere with correct segmentation. Each child AOI represents a 256x256 pixel area with 5 square meters per pixel. To achieve this, we calculated the correct maximum latitude and longitude coordinates for each child AOI based on the Ellipsoid Earth Model, ensuring each child AOI also covered a square area on the Earth's surface.

The subdivision process involved determining the appropriate coordinates for the child AOIs without simply splitting an already downloaded 1024x1024 pixel image. This approach avoids introducing resolution noise, which can occur when downscaling images. After initializing the 16 child AOIs, we collected images for each child AOI with a resolution of 256x256 pixels and 5 square meters per pixel. This process was repeated for each given year, oldest and most present.

Segmentation and After-Process

Consequently, we obtained 16 images for each AOI for every year under consideration. Using the trained Refined U-Net Model, we performed segmentation on each of these images. To refine the segmentation results, we applied a binary thresholding technique. Specifically, we utilized the Binary Thresholding method, after experimenting with Otsu vs Binary Thresholding methods.

Binary thresholding is a straightforward binarization technique that applies a fixed threshold value to an image for segmentation purposes. Unlike Otsu Thresholding, which automatically determines an optimal threshold value by minimizing intra-class intensity variance, binary thresholding uses a predefined threshold, such as 127, to separate the foreground from the background. Although Otsu Thresholding is advantageous for its adaptability to varying lighting conditions and image characteristics, it can sometimes introduce noise in certain examples. By using a binary threshold at 127, we achieved more consistent and reliable binary segmentation for each image, which is crucial for accurately delineating shorelines and calculating shoreline changes. This approach ensured that our segmentation process was stable and less susceptible to the noise that Otsu Thresholding occasionally added.

Extracting Shorelines

Afterwards, we utilized the Canny edge detection method to extract the shoreline of the AOI (Canny, 1986). The Canny edge detection algorithm is a multi-stage process that involves noise reduction, gradient calculation, non-maximum suppression, and edge tracking by hysteresis. This method was chosen due to its robustness in detecting a wide range of edges in images and its ability to produce thin and well-connected edge segments. The noise reduction step, achieved through Gaussian filtering, is crucial for reducing the impact of noise on the edge detection process. The gradient calculation, using Sobel operators, helps in identifying areas of rapid intensity change, which are indicative of edges. Non-maximum suppression then ensures that only the most significant edges are retained by thinning the edges to a single pixel width. Finally, edge tracking by hysteresis helps in distinguishing true edges from noise and other artifacts by using two thresholds.

The Canny edge detection method was particularly suitable for our study because it effectively highlights the discontinuities in intensity which correspond to the boundaries of the shoreline. Once the edges were extracted, we connected the resulting images to produce a mosaic of the parent area of interest. This mosaic provides a comprehensive and continuous view of the shoreline, enhancing the visibility and analysis of the region.

Calculate Shoreline Fluctuations

To accurately measure the changes in the shoreline within our AOI, we calculated the square meters of shoreline fluctuation, which includes both erosion and accretion. Shoreline fluctuations refer to the dynamic changes in the boundary between land and water, as discussed in detail on page 3 of this document. Our methodology for calculating shoreline changes involves processing the segmented images of the Refined U-Net Model, from two different time points: the past year and the most recent year that was given. Each pixel in these images represents an area of 5 square meters. To determine the changes, we subtracted the segmented images from one another. This subtraction allowed us to identify the remaining pixels that indicate either erosion or accretion. If there are no remaining pixels the land has not suffered from erosion or accretion respectively.

Erosion occurs when the land area in the most recent year is less than the land area in the past year. Here, the land (represented by white pixels with an sRGB value of 255) has been lost over time. To quantify erosion, we subtract the older image (with more land) from the most recent image (with less land). The remaining pixels in this subtraction represent the land that has been lost. We obtain the total square meters of land erosion by counting these remaining pixels and multiplying them by the resolution (5 square meters per pixel).

Accretion occurs when the land area in the most recent year exceeds the land area in the past year, indicating that water has receded, and land has been gained. We subtract the most present image (with more land) from the older image (with less land) to measure accretion. The remaining pixels in this case represent the newly gained land area. Again, we calculate the total square meters of land accretion by counting these pixels and converting them based on the resolution.

Case Studies of AOIs: Visualizing Coastal Changes

To test the main flow of our thesis project, we selected several AOIs based on two ESAs Articles “Monitoring coastal changes in Greece” (Agency, Monitoring coastal changes in Greece, 2024) and “Karavasta Lagoon, Albania” (Agency, Karavasta Lagoon, Albania, 2024) and generated comprehensive visualizations of coastal changes. We emphasized that Kalamaki Coast belongs to Natura 2000 (Natura 2000 (EEA), 2024). For visualization purposes, we have connected the 16 child images to produce the Parent AOI image with dimensions of 1024x1024 pixels. To illustrate changes over time, we marked the eroded pixels in red and the accreted pixels in green, overlaying these changes on the parent AOI as shown in *Figure 12*. This enhanced visualization allows for a clear and immediate understanding of the spatial distribution of erosion and accretion within the AOI.

Area of Interest: Kalamaki Coast, Zakynthos, Greece from 2017 to 2023

Figure 12: Coastal Fluctuations Visualization of Kalamaki Coast of Zakynthos, Greece from 2017 to 2023

```
Sea Erosion: 4695.880798339265 square meters
Sea Level Rise: 20132.339858646603 square meters
Absolute Difference: 23554.3381344547 square meters
```

Figure 13: Coastal Fluctuations in Square Meters of Kalamaki coast, Zakynthos, Greece from 2017 to 2023

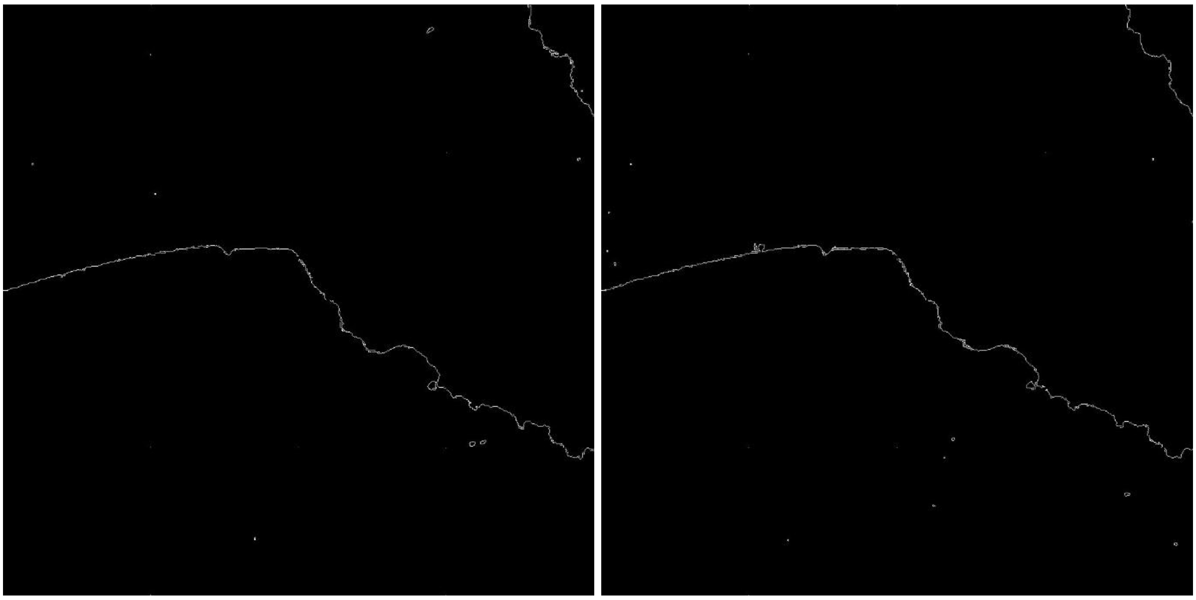


Figure 14: Detected Shorelines of Kalamaki Coast of Zakynthos, Greece in 2017 (left) and 2023 (right)

Figure 15: Satellite Images of Kalamaki Coast, Zakynthos, Greece in 2017 (left) and 2023 (right)

Area of Interest: Kiani Coast, Thessaloniki, Greece from 2017 to 2021

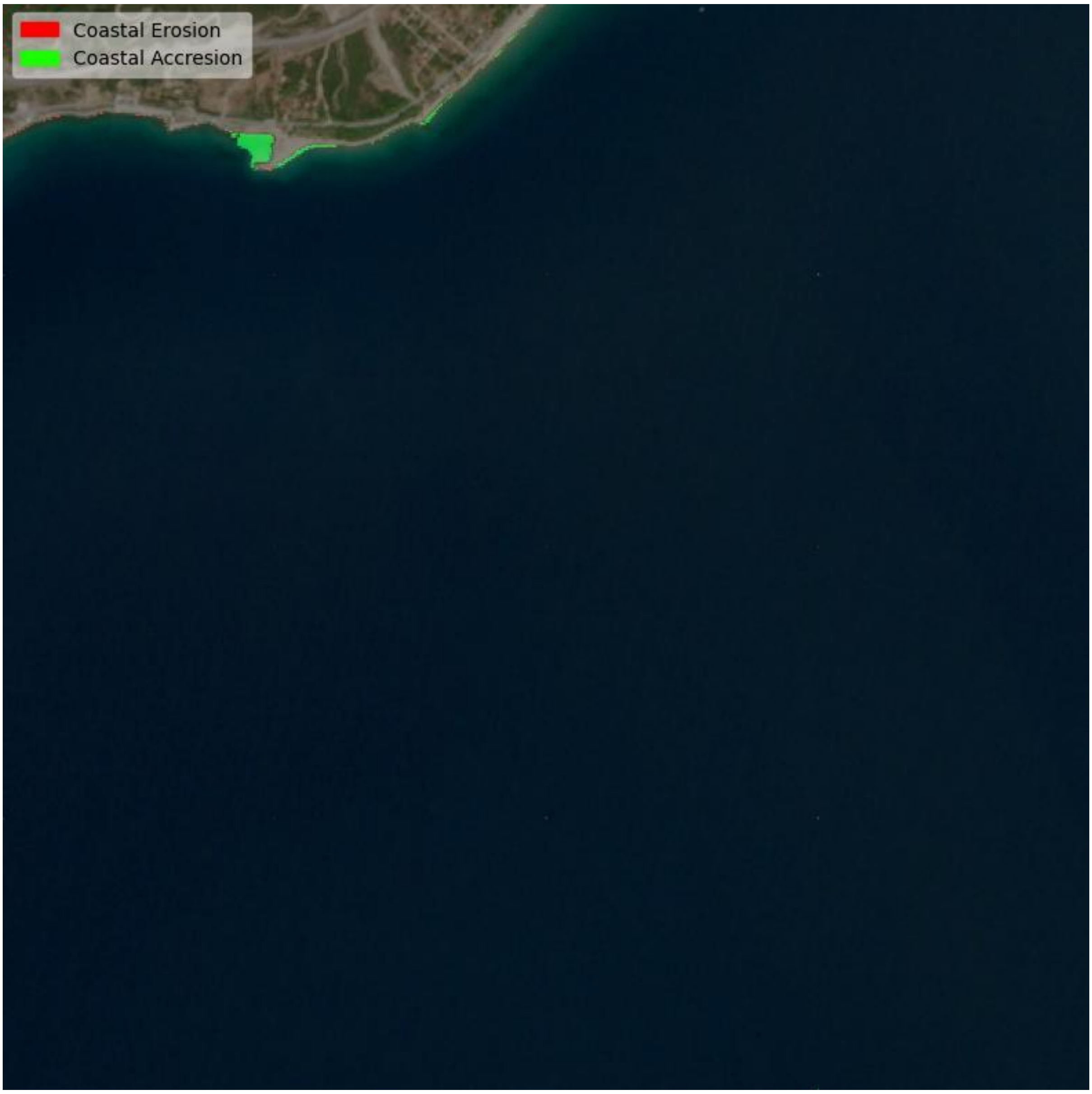


Figure 16: Coastal Fluctuations Visualization of Kiani Coast of Thessaloniki, Greece from 2017 to 2021

```
Coastal Erosion: 1622.510367230845 square meters
Coastal Accretion: 16324.968178479097 square meters
Absolute Difference: 17498.16834016458 square meters
```

Figure 17: Coastal Fluctuations in Square Meters of Kiani Coast, Thessaloniki, Greece from 2017 to 2021

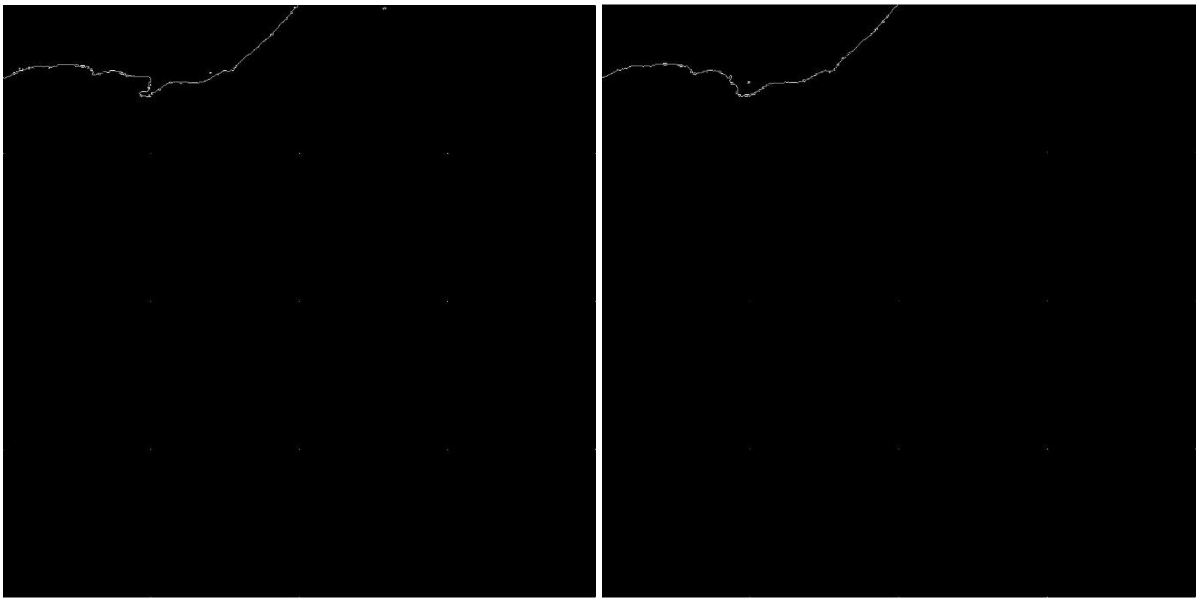


Figure 18: Detected Shorelines of Kiani Coast of Thessaloniki, Greece in 2017 (left) and 2021 (right)

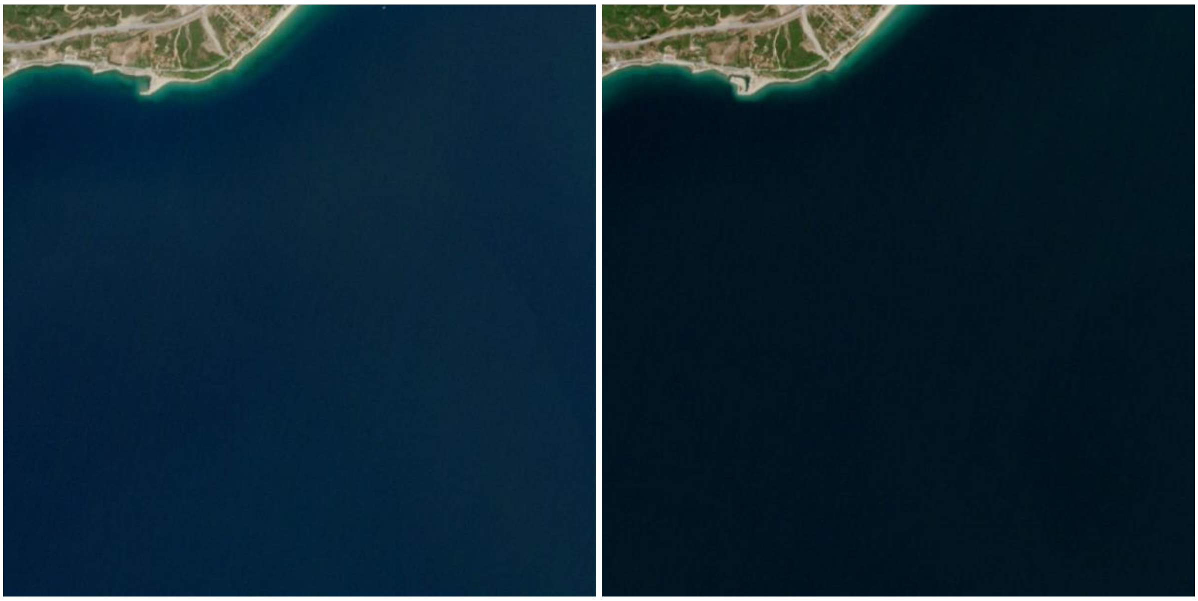


Figure 19: Satellite Images of Kiani Coast, Thessaloniki, Greece in 2017 (left) and 2021 (right)

Area of Interest: Adriatic Coast, Albania from 2016 to 2023

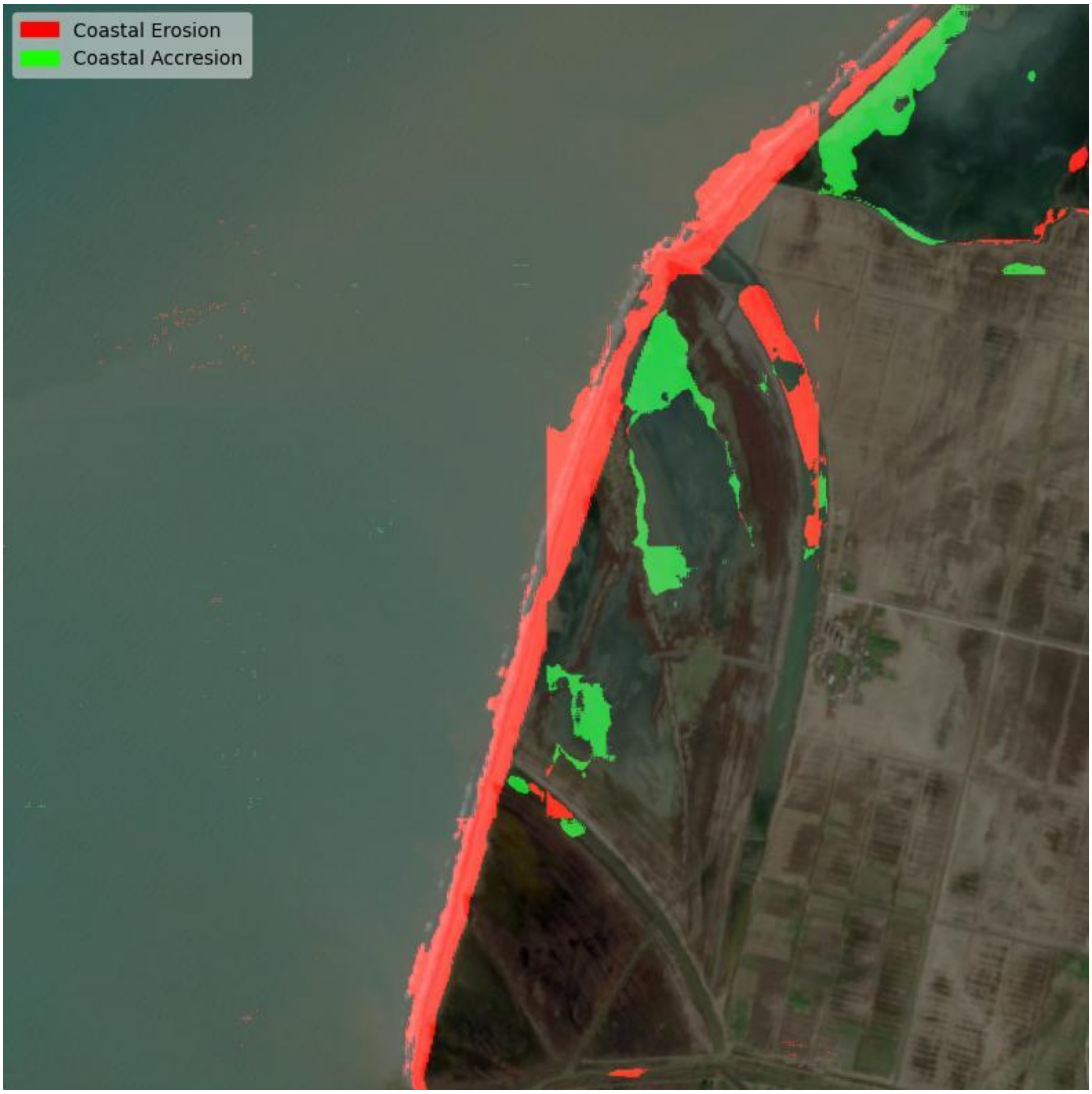


Figure 20: Coastal Fluctuations Visualization of Adriatic Coast, Albania from 2017 to 2023

```
Sea Erosion: 322545.6114513086 square meters
Sea Accretion: 786333.89181426 square meters
Absolute Difference: 1108205.242634981 square meters
```

Figure 21: Coastal Fluctuations in Square Meters of Adriatic Coast, Albania from 2017 to 2023

Figure 22: Detected Shorelines of the Adriatic Coast, Albania in 2017 (left) and 2023 (right)

Figure 23: Satellite Images of the Adriatic Coast, Albania in 2017 (left) and 2023 (right)

Analysis of Results and Identified Issues

As observed in the previous examples, the results vary in accuracy, sometimes showing noise. In the first example (*Figure 12*), we notice slight noise in the detected coastal fluctuations, with false erosion and accretion pixels appearing in the water body. Similarly, in the third example (*Figure 20*), although the correct erosion fluctuation on the shore is detected, there are false accretion and erosion detections within the lagoon on the shore. In the second example (*Figure 16*), the shore's accretion is correctly identified, although small red pixels, not visible in the visualization, contribute to the accretion measurement being almost 1622 square meters.

We can compare the model's results with those from the Enhanced NDWI Evalscript used to collect masks for training the network (*page* ***Error! Bookmark not defined.***). In Figure 26 and we can verify that the model had accurate results. However, *Figures 26* and *27* show that the model introduced noise, resulting in inaccurate outcomes.

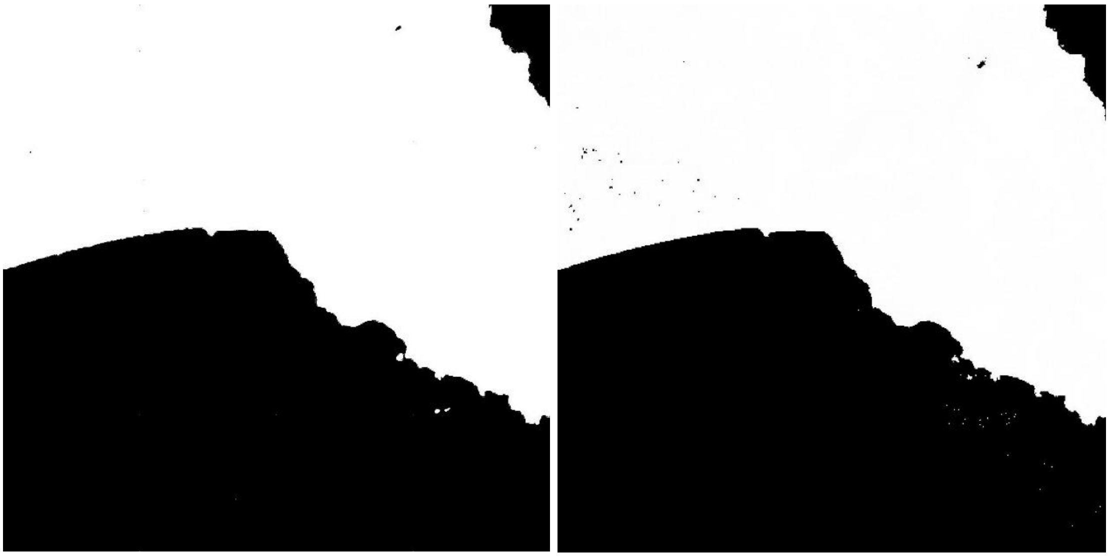


Figure 24: Segmentation (left) vs Enhanced NDWI Evalscript (right) of Kalamaki Coast, Zakynthos, Greece in 2017

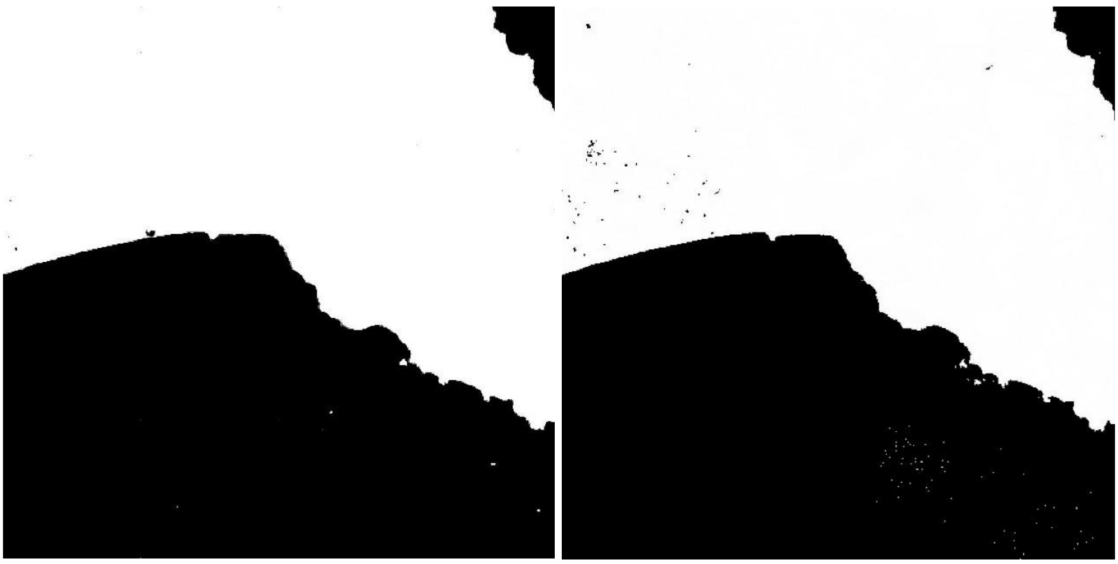


Figure 25: Segmentation (left) vs Enhanced NDWI Evalscript (right) of Kalamaki Coast, Zakynthos, Greece in 2023

Figure 26: Segmentation (left) vs Enhanced NDWI Evalscript (right) of Kiani Coast of Thessaloniki, Greece in 2017

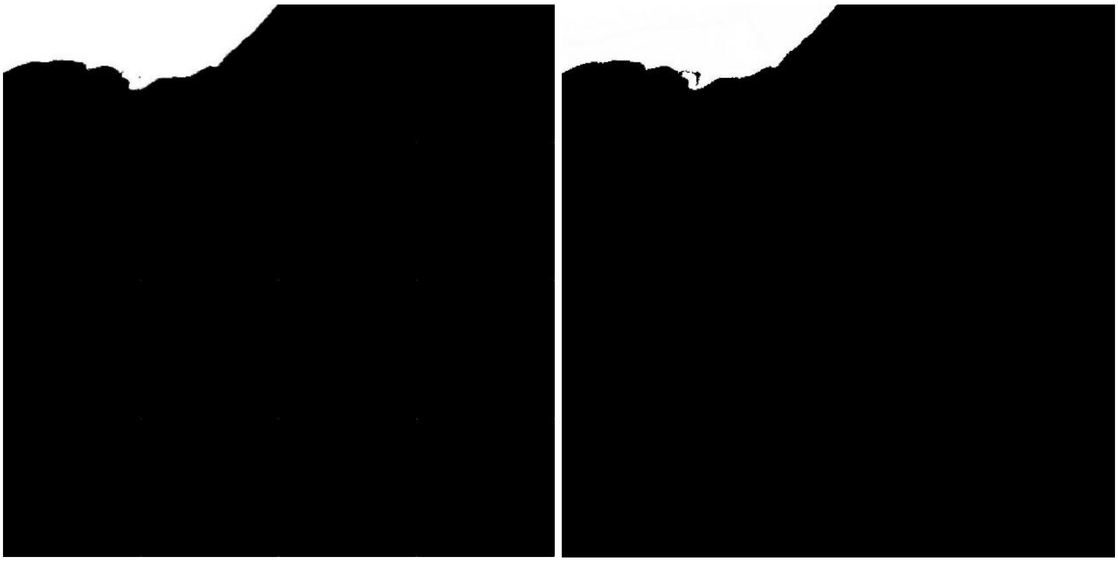


Figure 27: Segmentation (left) vs Enhanced NDWI Evalscript (right) of Kiani Coast of Thessaloniki, Greece in 2023

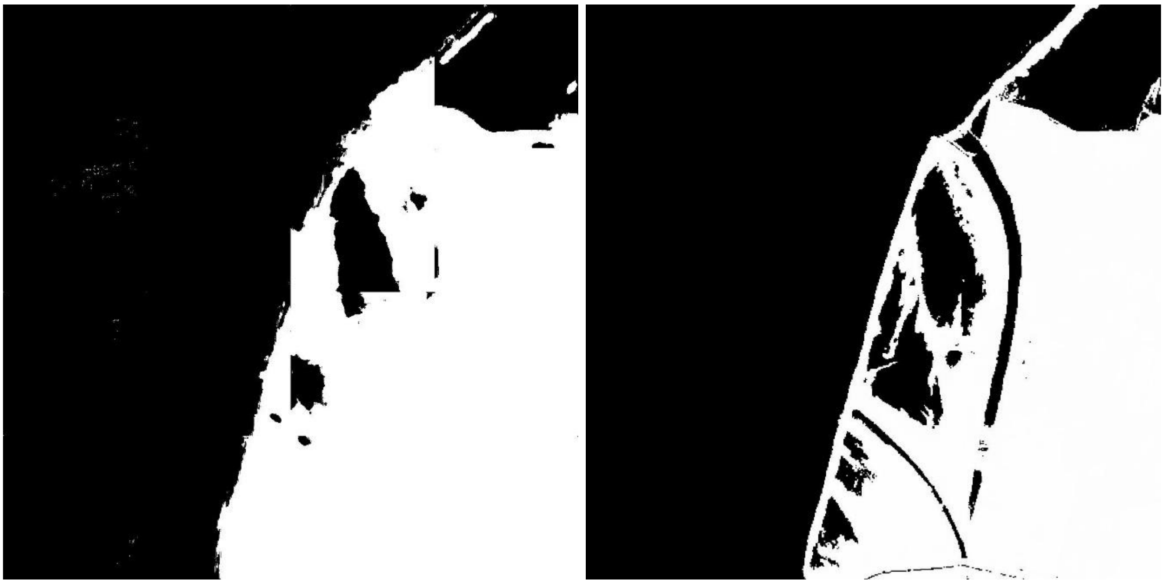


Figure 28: Segmentation (left) vs Enhanced NDWI Evalscript (right) of the Adriatic Coast, Albania in 2017

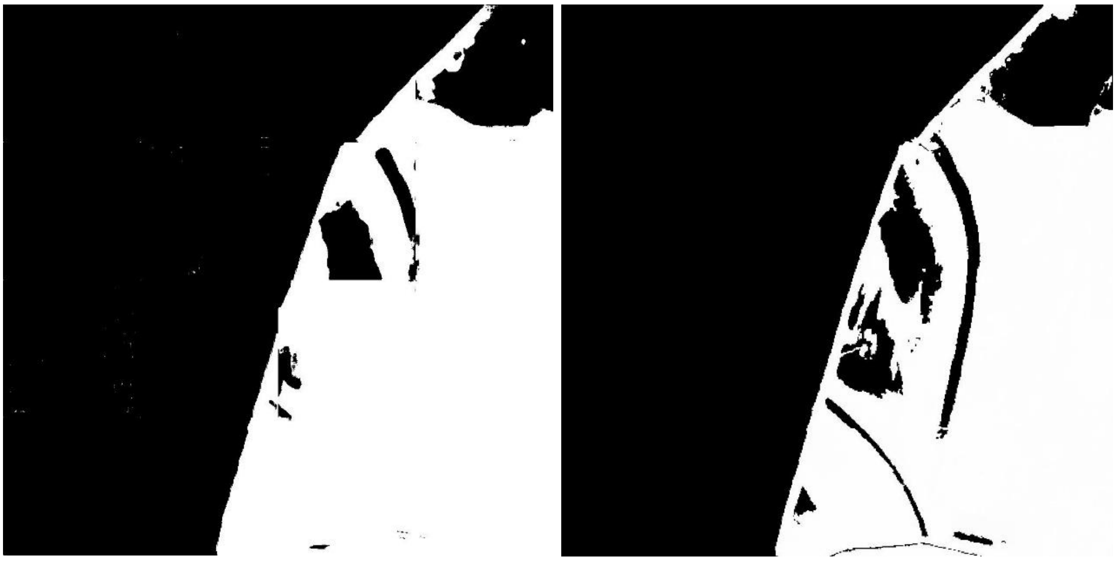


Figure 29: Segmentation (left) vs Enhanced NDWI Evalscript (right) of the Adriatic Coast, Albania in 2023

The likely reason for this noise is that the dataset used to train the model did not include enough examples of lagoons. Lagoons have unique colors and features that are different from other water bodies. Because of this, the artificial neural network (ANN) has trouble recognizing lagoons correctly. This leads to confusing results, with parts of the lagoon being wrongly identified as water and other parts as land.

Future Improvements

To address the issues identified, future improvements should focus on enhancing the dataset to include a wider variety of water bodies, especially lagoons. The collected dataset should include images from coasts all over the world, featuring a variety of shorelines and lagoons.

The current dataset of 8,000 images from just some Countries, may not be sufficient or diverse enough. It may contain too many images with just water or just land, which does not help the model learn to identify complex environments accurately. By expanding and diversifying the dataset, we can enhance the model's capability to precisely segment various types of coastal environments and minimize noise in the results.

This will help the artificial neural network (ANN) better recognize and differentiate between different environments. Additionally, refining the model's training process to account for these diverse examples of the more improved dataset, can improve accuracy and reduce noise in the segmentation results.

One effective method for improvement could be to check and normalize the input’s image HSV (Hue, Saturation, Value) values based on the mean HSV values of the dataset used for the training of the model. By normalizing the HSV values, the model can better handle variations in colour shades across different images, leading to more consistent and accurate segmentation. This normalization process would adjust the color values of the input images to match the average color characteristics of the training dataset, making it easier for the ANN to identify and classify water bodies and land accurately.

Another potential improvement is to enhance data augmentation techniques to simulate different environmental conditions, such as changes in lighting or seasonal variations. While we have already applied augmentation for shape variations, focusing on environmental factors can further improve the model's robustness. Techniques such as adjusting brightness and contrast, applying random noise, and simulating different weather conditions such as fog and rain can help the model better adapt to various real-world scenarios. This approach ensures the model is exposed to a broader range of conditions, improving its ability to accurately detect and segment water bodies and land in different environments.

Additionally, implementing a post-processing step to refine the model's predictions can further enhance accuracy. This could involve applying morphological operations, such as erosion and dilation, to remove small noise artifacts and smooth the segmentation boundaries. For noise removal, erosion can eliminate small, isolated pixels incorrectly classified as part of an object, and subsequent dilation can restore the size of the main objects, preserving their overall shape and size. For boundary smoothing, erosion can remove irregularities and small protrusions from the edges of the segmented regions. At the same time, dilation can then expand the eroded regions back to their original size, resulting in smoother boundaries and reduced noise.

Another option is to use conditional random fields (CRFs) to improve the spatial coherence of the segmentation by considering the relationships between neighbouring pixels. CRFs are probabilistic models that can consider the context and spatial arrangement of pixels in an image. Integrating CRFs into the segmentation process allows you to refine the boundaries and make the segmentation results more consistent and accurate. CRFs work by modelling the likelihood of a pixel belonging to a particular class based on the labels of its neighbouring pixels, thus enforcing spatial consistency and reducing classification errors at object boundaries. This technique can also significantly enhance the quality of the segmentation, especially in complex scenes with varying textures and colors.

A further improvement involves integrating additional data sources on the training of the model, such as multispectral and temporal satellite imagery, to refine the model. Multispectral imagery captures data at different wavelengths across the electromagnetic spectrum, providing more detailed information about the surface materials and conditions. By including bands beyond the visible spectrum, such as infrared, the model can gain insights into vegetation health, water content, and other features not discernible in regular RGB images. This additional information can help the model distinguish between different land cover types more accurately.

Regarding the Canny Edge Detection described in *page 32*, several advanced techniques could be explored for future innovations. Deep learning-based algorithms, such as Holistically-Nested Edge Detection (HED) networks, offer more accurate and context-aware edge detection. Another potential method is the use of active contour models (snakes), which evolve to fit the edges more precisely. Additionally, integrating machine learning techniques like Random Forest or Support Vector Machines (SVM) with traditional edge detection methods could enhance the accuracy and reliability of shoreline extraction.

Last but not least, calculating the differences between the model’s segmentation result and the NDWI result of the Sentinel-2 satellite is going to be an angular improvement for a better understanding of the accuracy of each result.

Conclusion

In this thesis, we explored the application of machine learning techniques to recognize and analyze shoreline and sea level changes using satellite imagery. The primary goal was to develop a robust model capable of accurately detecting these changes, which are crucial for environmental monitoring, coastal management, and mitigating the impacts of climate change.

Our study involved several key steps such as data collection, preprocessing, model training, and model evaluation. Satellite images from various sources were collected and provided the necessary data, which were then preprocessed to enhance quality and ensure consistency. We experimented with multiple U-Net convolutional neural networks (CNNs) Neural Network Architectures, to identify the most effective approach for our specific use case. Our best-performing model achieved a dice loss rate of 0.0062 and a dice coefficient of 0.993, indicating a strong potential for practical applications. Furthermore, we identified specific areas where our model could be improved, such as incorporating additional data sources and refining preprocessing and after-processing techniques.

The implications of our findings are significant. Accurate monitoring of shoreline and sea level changes can inform policymakers and help in developing strategies to protect coastal areas. It also contributes to our understanding of the broader impacts of climate change, providing valuable insights for future research and action. Future work could focus on integrating real-time data feeds and enhancing the computational efficiency of the models. Additionally, expanding the geographic scope of the study to include different coastal regions worldwide could enhance the model's generalizability and robustness.

In conclusion, this thesis has demonstrated the feasibility and effectiveness of using machine learning for shoreline and sea level change recognition. It also contributes to the scientific understanding of coastal processes and underscores the importance of using cutting-edge technology to address global environmental challenges. The integration of advanced machine learning with satellite remote sensing offers a powerful tool for monitoring and managing coastal zones. The advancements made in this study lay the groundwork for more comprehensive and scalable solutions in environmental monitoring and management.

References

A.D.Dongare, R.R.Kharde, & D.Kachare, A. (2012). Introduction to Artificial Neural Network. *International Journal of Engineering and Innovative Technolog*. Retrieved from https://api.semanticscholar.org/CorpusID:212457035

Abraham, A. (2005). Artificial Neural Networks. *Handbook of Measuring System Design*, 902-908.

Abraham, A. (2005). Nature and Scope of AI Techniques. *Handbook of Measuring System Design*, 894-900.

Agency, E. S. (2024, 06 05). *Karavasta Lagoon, Albania*. Retrieved from ESA: https://www.esa.int/ESA_Multimedia/Images/2017/03/Karavasta_Lagoon_Albania

Agency, E. S. (2024, 06 04). *Monitoring coastal changes in Greece*. Retrieved from ESA: https://www.esa.int/Applications/Observing_the_Earth/Space_for_our_climate/Monitoring_coastal_changes_in_Greece

Aggarwal, C. C. (2018). Recurrent Neural Networks. In C. C. Aggarwal, *Neural Networks and Deep Learning* (pp. 355-410). Springer INternational Publishing AG. doi:ISBN: 3030068560

Bharathvaj, S. A., & Salghuna, N. N. (2015). *Aquatic Procedia, 4*, 317-324. doi:https://doi.org/10.1016/j.aqpro.2015.02.043

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 679 - 698. Retrieved from https://ieeexplore.ieee.org/document/4767851

Charles, R., & O'Neill, J. (1985). A guide to coastal erosion processes. Retrieved from https://repository.library.noaa.gov/view/noaa/46278

Chuvieco, E. (2020). *Fundamentals of satellite remote sensing: An environmental approach.* doi:ISBN: 0429506481

Conway, E. D. (1997). *An introduction to satellite image interpretation.*

Copernicus. (2024). *About Copernicus*. Retrieved from Copernicus Web Site: https://www.copernicus.eu/en/about-copernicus

Copernicus Sentinel-2 (processed by ESA). (2021). *MSI Level-2A BOA Reflectance Product*. doi:https://doi.org/10.5270/S2_-znk9xsj

Copernicus, (ESA). (2024, 04 12). *Copernicus Data Space Ecosystem*. Retrieved from https://dataspace.copernicus.eu/

da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H., & dos Reis Alves, S. F. (2017). Artificial Neural Network Architectures and Training Processes. In *Artificial Neural Networks : A Practical Course* (pp. 21-28). doi:https://doi.org/10.1007/978-3-319-43162-8_2

Dickson, P. (2001). *Sputnik: The shock of the century.* Bloomsbury Publishing USA.

Ding, B. Q. (2018). Activation functions and their characteristics in deep neural networks. *2018 Chinese Control And Decision Conference (CCDC)*, (pp. 1836-1841). doi:10.1109/CCDC.2018.8407425

Docker Inc. (2024). *Docker*. Retrieved from https://www.docker.com/

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In *Machine Learning in Radiation Oncology: Theory and Applications* (pp. 3-11). doi:10.1007/978-3-319-18305-3_1

European Environment Agency. (2024, 04 10). *Global and European sea level rise*. Retrieved from Europa: https://www.eea.europa.eu/en/analysis/indicators/global-and-european-sea-level-rise

European Space Agency. (2024). *About ESA*. Retrieved from The European Space Agency Web Site: https://www.esa.int/About_Us

European Space Agency. (2024). *Copernicus Programm*. Retrieved from SentiWiki: https://sentiwiki.copernicus.eu/web/copernicus-programme

European Space Agency. (2024). *ESA Missions*. Retrieved from The European Space Agency Web Site: https://www.esa.int/Missions

European Space Agency. (2024). *Senitnel-2 Scquisition Resolution*. Retrieved from SentiWiki: https://sentiwiki.copernicus.eu/web/s2-mission#S2Mission-AcquisitionResolutionsS2-Mission-Acquisition-Resolutionstrue
European Space Agency. (2024). *Sentinel-1*. Retrieved from SentiWiki: https://sentiwiki.copernicus.eu/web/sentinel-1
European Space Agency. (2024). *Sentinel-2*. Retrieved from SentiWiki: https://sentiwiki.copernicus.eu/web/sentinel-2
European Space Agency. (2024). *Sentinel-2 Applications*. Retrieved from SentiWiki: https://sentiwiki.copernicus.eu/web/s2-applications
European Space Agency. (2024). *Sentinel-3*. Retrieved from SentiWiki: https://sentiwiki.copernicus.eu/web/sentinel-3
European Space Agency. (2024). *Sentinel-5P*. Retrieved from SentiWiki: https://sentiwiki.copernicus.eu/web/sentinel-5p
Fox-Kemper, B. (n.d.). Ocean, Cryosphere and Sea Level Change. *AGU Fall Meeting Abstracts*, (pp. U13B-09). Retrieved from https://ui.adsabs.harvard.edu/abs/2021AGUFM.U13B..09F
Goodfellow, I., Bengio, Y., & Courville, A. (2016). Chapter 5.2: Capacity, Overfitting and Underfitting. In I. Goodfellow, Y. Bengio, & A. Courville, *Deep learning* (pp. 107-116). Cambridge, London: Masachusetts Institute of Technolog. Retrieved from https://books.google.gr/books?hl=en&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&dq=Goodfellow,+I.,+Bengio,+Y.,+%26+Courville,+A.+(2016).+Deep+Learning.+MIT+Press.+Chapter+5+discusses+overfitting+in+neural+networks+and+regularization+techniques.&ots=MOO-coqHTX&sig=J
Hecht-Nielsen, R. (1989). *Neurocomputing.* Addison-Wesley Longman Publishing Co.
Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 107-116. doi:10.1142/S0218488598000094
Hosseini, M.-P., Lu, S., Kamaraj, K., Slowikowski, A., & Venkatesh, H. C. (2020). Deep Learning Architectures. In *Deep Learning: Concepts and Architectures* (p. 3). doi:https://doi.org/10.1007/978-3-030-31756-0_1
Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning* (pp. 448-456). 448: PMLR. Retrieved from https://proceedings.mlr.press/v37/ioffe15.html
Kuleli, T. (2010). Quantitative analysis of shoreline changes at the Mediterranean Coast in Turkey. *Environmental Monitoring and Assessment*, 387-387.
Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 11-26. doi:https://doi.org/10.1016/j.neucom.2016.12.038
Ludwig, P. E., Reddy, V., & Varacallo, M. (2024). Neuroanatomy, Neurons. *StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing*. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK441977/
Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models., (p. 3). Atlanta, Georgia, USA. Retrieved from http://robotics.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf
Mentaschi, L., Vousdoukas, M. I., Pekel, J.-F., Voukouvalas, E., & Feyen, L. (2018). Global long-term observations of coastal erosion and accretion. *Scientific Reports*, 12876. doi:https://doi.org/10.1038/s41598-018-30904-w
Mertikas, S., Partsinevelos, P., Mavrocordatos, C., & Maximenko, N. (2021). Chapter 3 - Environmental applications of remote sensing. In *Pollution Assessment for Sustainable Practices in Applied Sciences and Engineering* (pp. 107 - 163). Butterworth-Heinemann.

Retrieved from https://www.sciencedirect.com/science/article/abs/pii/B9780128095829000037
National Aeronautics and Space Administration, N. (2024). *The Electromagnetic Spectrum*. Retrieved from NASA Science: https://science.nasa.gov/ems
Natura 2000 (EEA). (2024, 06 04). Retrieved from https://natura2000.eea.europa.eu/?sitecode=GR2210002&views=Sites_View
NVIDIA. (2024). *CUDA*. Retrieved from https://developer.nvidia.com/cuda-zone
Office of Geomatics, N. G.-I. (2024, 04 02). *WGS84*. Retrieved from NGA Geomatics: https://earth-info.nga.mil/?dir=wgs84&action=wgs84
Pianca, C., Holman, R., & Siegle, E. (2015). Shoreline variability from days to decades: Results of long-term video imaging. *Journal of Geophysical Research: Oceans, 120*(3), 2159-2178. doi:https://doi.org/10.1002/2014JC010329
Rajasree, B. R., Deo, M. C., & Sheela Nair, L. (2016). Effect of climate change on shoreline shifts at a straight and continuous coast. *Estuarine, Coastal and Shelf Science*, 221-234. doi:https://doi.org/10.1016/j.ecss.2016.10.034
Roddy, D., Jones, W. L., & Long, D. G. (2006). *Satellite communications.* McGraw-hill New York.
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015.* Cham: Springer International Publishing. Retrieved from https://arxiv.org/abs/1505.04597
Rumelhart, D. E. (1986). Learning representations by back-propagating errors. *Nature*, 533-536. Retrieved from https://doi.org/10.1038/323533a0
Salghuna, N. N., & Bharathvaj, S. A. (2015). Shoreline Change Analysis for Northern Part of the Coromandel Coast. *Aquatic Procedia*, 317-324.
Schaul, T., Borsa, D., Modayil, J., & Pascanu, R. (2019). Ray interference: a source of plateaus in deep reinforcement learning. *arXiv preprint arXiv:1904.11455*. Retrieved from https://arxiv.org/abs/1904.11455
Sentinel-Hub. (2024). Retrieved from https://docs.sentinel-hub.com/api/latest/#how-to-read-this-documentation
Sentinel-Hub. (2024). *custom-scripts.sentinel-hub*. Retrieved from custom-scripts.sentinel-hub: https://custom-scripts.sentinel-hub.com/sentinel-2/l2a_optimized/
Sentinel-Hub. (2024). *custom-scripts.sentinel-hub*. Retrieved from custom-scripts.sentinel-hub: https://custom-scripts.sentinel-hub.com/sentinel-2/ndwi/
Sentinel-Hub. (2024). *Evalscript*. Retrieved from https://docs.sentinel-hub.com/api/latest/evalscript/
Sentinel-Hub. (2024). *True Color*. Retrieved from https://custom-scripts.sentinel-hub.com/sentinel-2/true_color/
Sinergise. (2017). *Python API*. Retrieved from Sentinel-Hub: https://sentinelhub-py.readthedocs.io/en/latest/
Sommervold, O., Gazzea, M., & Arghandeh, R. (2023). A Survey on SAR and Optical Satellite Image Registration. *Remote Sensing, 15*(3). doi:10.3390/rs15030850
Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (pp. 240-248). Cham: Springer International Publishing. Retrieved from https://www.mendeley.com/reference-manager/library/all-references
Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning. *Journal of Cognitive Neuroscience*, 126-134. Retrieved from https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=reinforcement+learning&btnG=&oq=Reinfor#d=gs_cit&t=1721043472684&u=%2Fscholar%3Fq%3Dinfo%3A_Z9ZEZFNmBwJ%3Ascholar.google.com%2F%26output%3Dcite%26scirp%3D5%26hl%3Den
Tensorflow. (2024). Retrieved from https://www.tensorflow.org/

Vecchio, A., Anzidei, M., & Serpelloni, E. (2024). Sea level rise projections up to 2150 in the northern Mediterranean coasts. *Environmental Research Letters*, 014050. doi:10.1088/1748-9326/ad127e
Vu-Quoc, P. L. (2019, March 17). *Neuron.svg.* Retrieved from English Wikipedia: https://commons.wikimedia.org/wiki/File:Neuron.svg
Wikipedia. (2024). *Earth Ellipsoid*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/Earth_ellipsoid
Wikipedia. (2024). *Earth Radius - Meridional*. Retrieved from https://en.wikipedia.org/wiki/Earth_radius#Meridional
Wikipedia. (2024). *Earth Radius - Prime Vertical*. Retrieved from https://en.wikipedia.org/wiki/Earth_radius#Prime_vertical
Wikipedia. (2024). *Earth_ellipsoid*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/Earth_ellipsoid
World Meteorological Organization. (2024). *OSCAR*. Retrieved from OSCAR: https://space.oscar.wmo.int/
World Meteorological Organization. (2024). *OSCAR - Satelite Programmes*. Retrieved from OSCAR: https://space.oscar.wmo.int/satelliteprogrammes
World Meteorological Organization. (2024). *OSCAR - Satellites*. Retrieved from OSCAR: https://space.oscar.wmo.int/satellites
World Meteorological Organization. (2024). *OSCAR - Space Agencies*. Retrieved from OSCAR: https://space.oscar.wmo.int/spaceagencies
Yann LeCun, Y. B. (2015). Deep learning. *Nature*.
Yegnanarayana, B. (2009). *Artificial Neural Networks.* doi:ISBN: 8120312538
Zakaria, M., Mabrouka, A. S., & Sarhan, S. (2014). Artificial Neural Network: A Brief Overview. *Neural Networks*, 2.