Jorge Miguel Soares de Campos Assunção

Licenciado em Engenharia Eletrotécnica e de Computadores

Terrain classification using machine learning algorithms in a multi-temporal approach A QGIS plug-in implementation

Dissertação para obtenção do Grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Orientador: José Manuel Fonseca, Professor Associado com Agregação, Universidade NOVA de Lisboa

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Outubro, 2021

Terrain classification using machine learning algorithms in a multi-temporal approach
A QGIS plug-in implementation

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

Firstly, I would like to extend my gratitude to my adviser, José Fonseca for the opportunity and trust in allowing me to participate in this endeavor, under his time and help in all stages of the development of this dissertation. In addition, I would like to thank the IPSTERS project team and the CA3 Computational Intelligence Research Group as a whole, for the valuable advice provided and the mutual help environment displayed throughout the entirety of my stay.

Secondly, I wish to acknowledge the invaluable help provided by my incredible group of coworkers and friends throughout all my academic years, culminating in the development of this document. Their continued help extends beyond schooling into character building, influence which I'll carry throughout my life. I am eternally indebted.

Lastly, I would like to express my gratitude towards my family for the unwavering support and patience shown throughout my academic process, especially during the most recent abnormal years. Thank you.

Abstract

Land cover and land use (LCLU) maps are essential for the successful administration of a nation's topography, however, conventional on-site data gathering methods are costly and time-consuming. By contrast, remote sensing data can be used to generate up-to-date maps regularly with the help of machine learning algorithms, in turn, allowing for the assessment of a region's dynamics throughout time.

The present dissertation will focus on the implementation of an automated land use and land cover classifier based on remote sensing imagery provided by the modern sentinel-2 satellite constellation. The project, with Portugal at its focus, will expand on previous approaches by utilizing temporal data as an input variable in order to harvest the contextual information contained in the vegetation cycles.

The pursued solution investigated the implementation of a 9-class classifier plug-in for an industry standard, open-source geographic information system. In the course of the testing procedure, various processing techniques and machine learning algorithms were evaluated in a multi-temporal approach. Resulting in a final overall accuracy of 65,9% across the targeted classes.

Keywords: Sentinel-2, Machine learning, Land cover and land use

Resumo

Mapas de uso e ocupação do solo são cruciais para o entendimento e administração da topografia de uma nação, no entanto, os métodos convencionais de aquisição local de dados são caros e demorados. Contrariamente, dados provenientes de métodos de sensoriamento remoto podem ser utilizados para gerar regularmente mapas atualizados com a ajuda de algoritmos de aprendizagem automática. Permitindo, por sua vez, a avaliação da dinâmica de uma região ao longo do tempo.

Utilizando como base imagens de sensoriamento remoto fornecidas pela recente constelação de satélites Sentinel-2, a presente dissertação concentra-se na implementação de um classificador de mapas de uso e ocupação do solo automatizado. O projeto, com foco em Portugal, irá procurar expandir abordagens anteriores através do aproveitamento de informação contextual contida nos ciclos vegetativos pela utilização de dados temporais adicionais.

A solução adotada investigou a produção e implementação de um classificador geral de 9 classes num plug-in de um sistema de informação geográfico de código aberto. Durante o processo de teste, diversas técnicas de processamento e múltiplos algoritmos de aprendizagem automática foram avaliados numa abordagem multi-temporal, culminando num resultado final de precisão geral de 65,9% nas classes avaliadas.

Palavras-chave: Sentinel-2, Aprendizagem automática, Cartografia de uso e ocupação do solo

Contents

List of Figures

List of Tables

Acronyms

2D Two dimensions

3D Three dimensions

AI Artificial intelligence

ANN Artificial neural network

API Application programming interface

AUC Area under curve

BSI Bare Soil Index

CNN Convolutional neural networks

COS *Carta de Uso e Ocupação do Solo*

CS Cluster sampling

DEM Digital elevation models

DGT *Direção-geral do Territorio*

DT Decision tree

DTM Digital terrain models

ESA European Space Agency

FCT Fundação para a Ciência e a Tecnologia

FN False negative

FP False positive

GIS Geographic information system

GLCM Grey level co-occurrence matrix

GUI Graphical user interface

IDE Integrated Development Environment

IPSTERS IPSentinel Terrestrial Enhanced Recognition System

KNN K-Nearest Neighbor

LCLU Land cover and land use

LiDAR Light Detection And Ranging

LTA Long Term Archive

LWIR Long-wave infrared

ML Machine learning

MLP Multilayer perceptron

MMU Minimum mapping unit

MNDWI Modified Normalized Difference Water Index

NDBI Normalized Difference Built-up Index

NDGI Normalized Difference Greenness Index

NDMI Normalized Difference Moisture Index

NDVI Normalized difference vegetation index

NIR Near-infrared

OA Overall accuracy

OBIA Object-based image analysis

PPV Positive predictive value

QGIS Quantum geographic information system

RBF Radial basis function

ReLU Rectified Linear Unit

RF Random forest

ROC Receiver operating characteristic

SAR Synthetic Aperture Radar

SGD Stochastic gradient descent

SRS Simple random sampling

SSUS Stratified systematic unaligned sampling

STRS Stratified random sampling

SVM Support vector machine

SWIR Short-wave infrared

TN True negative

TP True positive

TPR True positive rate

Chapter 1

Introduction

Land cover and land use (LCLU) maps are essential for the successful administration of a nation's topography, however, the traditional on-site data gathering methods are expensive and can take considerable amounts of time to produce. By contrast, remote sensing data can be used to generate up-to-date maps regularly, allowing the accurately assessment of region dynamics throughout time [1]. In recent years, the increase in the availability of high-resolution, open-access satellite images has allowed the application of automatic machine learning methods to the generation of LCLU maps. Machine learning methods have also benefited from the advances in remote sensing, the capability of these algorithms to learn complex, non-linear, patterns is improved with the availability of extensive data, this has led to a considerable amount of attention being given to machine learning classification over the last decades [2, 3]. Furthermore, with the increase in feature complexity and uneven statistical distributions in the available data, previous research show that traditional statistical classifiers(e.g maximum likelihood, minimum distance) have been surpassed by machine learning algorithms in accuracy and reliability, especially in tasks with abundant training data [4], in turn, increasing the use of these classifiers in land cover and land use mapping[5, 6]. Over the years a large number of applications have been implemented around such LCLU maps, granting extensive geographical coverage for: management of natural resources, global change studies [7], urban planning, conservation [8, 9]. With the recent focus on environmental sustainability, adaptation the land management in response to climate change is of special importance, aiming to anticipate the changes in climate conditions and mitigate their potential negative consequences[10]. Since changes in forest and environmental management are likely to take part in carbon emissions and sequestration [11], LCLU maps can also help measuring the effects and progress of environmental health restoration by monitoring changes in vegetation communities [12].

1.1 Dissertation structure

The present document will be organized in the following structure.

- **Chapter 1:** Establishes the dissertation's subject through the introduction of the thesis motivation, problem and proposed solution.
- **Chapter 2:** Establishes the knowledge foundation through the introduction of basic concepts and technology related with remote sensing applications. Previous work in the field is also presented along with a general comparison of the reviewed literature.
- **Chapter 3:** Expands on the proposed solution through the explanation of the strategies and methodologies necessary for the reliable comparison of the employed procedures.
- **Chapter 4:** Provides the dissertation's findings along with the material implementation of the proposed strategy.
- **Chapter 5:** Presents the conclusion of the overall thesis motivation, work, and results, as well as proposed future developments.

1.2 Problem statement

Autonomous land use and land cover classifiers are strong tools to analyse and manage environmental resources and human expansion across the world in a fast and reliable way. The performance of these classifiers, however, is highly dependent on the methods used and the regions used to gather data, usually presenting a dip in accuracy when applied to different regions. Phenomenon potentially attributed to the possible distinct characteristics between elements from the same classes across thematic regions, be it from the dissimilarity between vegetation species or different regional class aggregation. Additionally, advances in remote sensing sources, such as the deployment of higher resolution sensors, and novel machine learning techniques able to fuse and utilize diverse complementary imagery data, further increase the need of constant experimentation and refining of these classifiers.

Portugal has been a site for large land cover and land use changes in recent years [13, 14], requiring constant updating of LCLU maps. However, traditional methods such as visual interpretation of aerial and satellite images, are costly and time-consuming, hampering the feasibility of regular production. Consequently, Portugal is a strong candidate for autonomous terrain classification technologies, allowing the production of crucial information for projection and modeling applications [15]. Over the years, academic research has been done to answer this necessity, however, the abundance of such projects is still limited. Furthermore, the available work is often times outdated, accomplished using primordial machine learning algorithms and low resolution satellite imagery [16]. In addition, studies are commonly implemented in isolated environments, hindering the possibility of future maintenance and upgradability, reducing the real-world applicability of such accomplishments.

1.3 Proposed solution

In this dissertation, the creation of a plugin for an open-source geographic information system (i.e. QGIS), is proposed in order to implement a Portugal focused LCLU classifier using machine learning algorithms.

The aforementioned classifier will employ multispectral imagery data from the recently deployed sentinel-2 satellite pair from the Copernicus Programme, to study and investigate the use the of contextual information contained in the vegetation cycles as input data. Throughout the training procedure, different classifier architectures shall be tested using open-source python packages, to access the benefits of a multi-temporal pixel-based approach. Additionally, the project will consider the practical aspect of the plug-in in order to allow for future development and maintenance, in turn, providing an additional reliable open-source option for LCLU map generation within Portugal. Future data and training may then be supplied in order to expand the employment range to other locations.

1.4 About the project

The present thesis subject is proposed as part of a multi-disciplinary project by the name IPSentinel Terrestrial Enhanced Recognition System (IPSTERS), created as a *Fundação para a Ciência e a Tecnologia (FCT)* project [17] in 2019 by the CA3 Computational intelligence Research Group[18]. The project's main focus, as a unit of the IPSentinel [19], is to explore the applications and limitations of artificial intelligence algorithms in the creation of level-3 products for land applications in environments with large amounts of remotely sensed data.

Chapter 2

Review of State-of-art

In this chapter, machine learning concepts will be introduced along with some of the technologies used in remote sensing applications, in particular, technology related with the implementation of the QGIS plug-in. Subsequently, a review on the state-of-art along with previously reported implementations will be presented.

2.1 Basic concept introduction

Machine learning (ML), first coined by Arthur Lee Samuel in 1959 [20], is usually referred to as an application of Artificial intelligence (AI) that allows a system to perform a specific task without being explicitly programmed with instructions. The system is designed to autonomously "learn"and improve from past experiences (i.e. data observations), allowing for the inference of patterns and rules that generalize the characteristics of the particular task. Although the term has been used for over 60 years, for some of its history it was regarded as unfeasible, fluctuating between accomplishments and complications related with practical and theoretical data acquisition and representation problems [21]. In recent decades, however, deep learning along with advances in hardware capability and the explosion of available data world-wide has propelled machine learning technologies and AI to a driving force in the conception of Industry 4.0 [22].

Machine learning techniques are vast and distinct, the implementation of which is defined by the available resources and the intended application. In the next sections a description of the methods associated with the implementation of autonomous learning algorithms will be presented along with the reported findings from the reviewed academic literature.

2.1.1 Supervised and Unsupervised learning

In the scope of remote sensing, learning algorithms may be separated in two categories, supervised learning and unsupervised learning. Among the learning methods, supervised learning is the most common technique [23]. The machine is fed training data along with the correct label, also called a pair, consisting of an input vector and an output value (or vector). By mapping inputs to outputs based on the provided pairs, the machine infers a function that can be used for mapping new unseen examples. Back propagation is an example of supervised learning, represents a widely used algorithm, where a function E, representing the error between the outputs of the network and a set of target outputs, is minimized in each iteration [1].

Unsupervised learning refers to a type of machine learning algorithm used to extract patterns and deduce rules from data without pre-existing labelled responses. Cluster analysis is the most common method of unsupervised learning, clusters are modeled to group elements from a data set with shared attributes by measuring the similarity between them using Euclidean and probabilistic distances [24]. The method reacts to new data by identifying the absence or presence of the inferred commonalities. Although the method has seen limited use in land cover and land use applications compared to supervised learning, studies have used it to identify sub classes, as well as multi-objective classification with Support vector machines in fuzzy clustering schemes, achieving satisfactory results [25, 26]. It is however, expected that unsupervised learning algorithms may surpass the use of supervised learning methods in future applications [23], the ability to perform tasks with fully unprocessed data allows it to be efficiently applied to most real-world applications, finding increase use in deep learning implementations. Occasionally, when the data set is too small to enforce supervised learning, a hybrid approach can be implemented, combining small amounts of labeled data with large amounts of unlabeled data, this technique is referred to as semi-supervised learning. The aforementioned technique, produces a larger training set by mapping the unlabeled data into clusters and assuming their respective classes based on the mapping of labeled samples [27]. The implementation of semi-supervised learning in remote sensing applications has generated satisfactory results [28], especially in classifiers based on Support vector machines [29].

2.1.2 General implementation process

In order to apply machine learning classifiers to remote sensing data, four steps are typically practiced. The first step usually consists in gathering data from images where all the classes are present, this data is then used to create a training set and a testing set, the data in these sets should be mutually exclusive to provide a more accurate result, mirroring the accuracy of the classifier when applied on unseen data [30]. The training data can later be treated for outliers and balanced across the instances of each class. Although the necessary amount of data used to train the classifier is dependent on the complexity of the system, extensive amounts are desirable in order to train an accurate classifier [5], with previous studies stating that the size of the training sample should account for 0.25% of the total study area [31, 32]. In order to generate an error matrix that represents the testing sample accurately we need to diminish the statistical variation present in the testing sample, this can be achieved by applying different sampling schemes, such as Stratified random sampling (STRS), Cluster sampling (CS), Simple random sampling (SRS) and Stratified systematic unaligned sampling (SSUS) [1].

The second step consists in selecting a suitable model. The choice of the classifier is in itself a decisive factor in the reliability of the generation of land cover and land use mapping [33].Factors such as, the available resources, the presence of noisy or incomplete input data, and desirable complexity of the system, should be considered. The chosen model should then be optimized in order accurately classify all classes while avoiding over-fitting, this involves feature polynomial exploration and hyper parameter determination [1]. The choice of hyper parameters impact the learning rate, fitting level and the structure of the machine learning classifiers [34], usually presenting a trade-off between the simplicity of the classifier and the achieved accuracy. In order to test different hyper parameter, researchers split the training set into training and cross-validation subsets. Allowing the classifier to compare the effect of the hyper parameters on the cross-validation subset, leaving the final test set "out of sight", this is done to prevent indirectly training the classifier with the test data, maintaining an unbiased result on unseen data.

In the third step, the classifier is presented with an entire image, it classifies every pixel to the perceived class. Some studies [35], approach the classification method differently than the regular hard classification, in hard (or crisp) classification, each pixel can only belong to a restricted class, in soft (or fuzzy) classification, each pixel can partially belong to multiple classes [36].

Lastly, the generated image is compared against a reference map, as seen in figure 2.1, and accessed classification performance the classifier. The metrics used to access the classification performance of a given classifier are abundant within the literature, being the two most commonly used: Overall accuracy (OA) and Cohen's Kappa coefficient [32]. In recent years however, Kappa statistic has seen a decrease in use in remote sensing classification [37], for presenting difficulty in interpretation and being considered an excessively conservative measure of agreement [38]. Overall accuracy, while commonly used, disregards the specific performance of each class in its final result, leading to a possibly deceiving result in highly unbalanced data sets [39]. Another accuracy assessment can be gathered by calculating the Area under curve (AUC) of a Receiver operating characteristic (ROC) curve graph. The aforementioned metric can be used in binary classification, measuring the quality of the model's prediction irrespectively of the chosen classification threshold, accessing the model per class distinguishability instead of their absolute values.

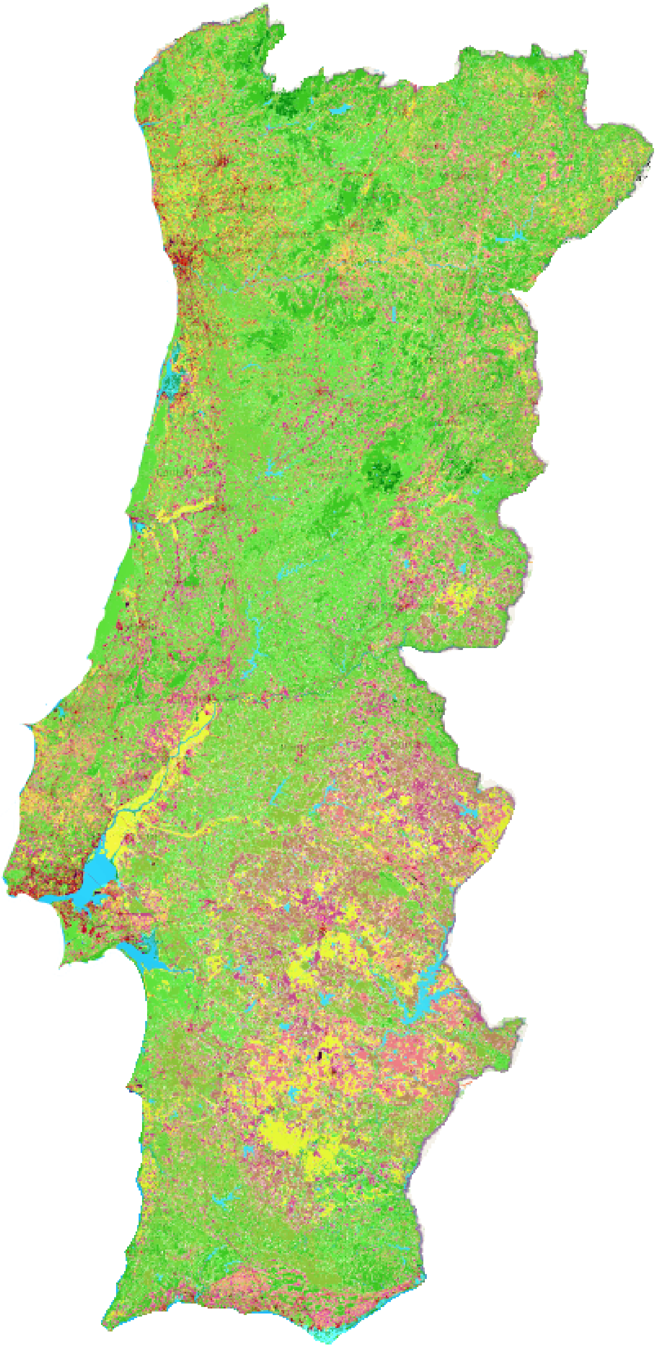


Figure 2.1: Reference LCLU map for Portugal (2018) [40].

2.1.3 Data fusion

In order to increase the performance of machine learning classifiers, input data from multiple sources can also be fused. Conventional, remote sensing data can be fused using three different levels of data fusion techniques, the pixel/data level, the feature level and the decision level [41]. The pixel/data level (also referred as low level) refers to the combination of raw data derived from multiple sources into a single resolution, this process is believed to increase the information over either input data, by revealing correlation between data samples [42]. This process, however, has a caveat when re-sampling data with different original resolution, where information is either lost or presumed depending on the scale disparity [43]. Feature level fusion (described as high level) operates by extracting various features (e.g. corners, lines, texture parameters, etc.) from the available data samples, the features are then combined into one or more features maps to be used as input data for further processing. Lastly, decision level fusion (i.e. such as voting, statistical and fuzzy logic based methods), computes the outcome from multiple algorithms before combining them to produce a final decision. If the algorithms results are expressed as well defined votes, the method is called hard classification, if the results are expressed as a confidence parameter the method takes the name of soft classification [42]. In order to further improve the classification performance of the decision level fusion technique, weighted decisions strategies may be used instead of relying solely in the majority vote. This method allows for the per-class accuracy of each algorithm to be considered, increasing the decision leverage of the higher accuracy algorithms. As concluded by [44], combined data sets produced better accuracy when using Random forest and Support vector machine classifiers over standalone variations. [45] have also evidenced that Random forest produces higher overall accuracy when mapping rice paddies using combined data from Sentiel-1A and Landsat-8.

2.1.4 Classification types

In remote sensing applications, two classification methods are usually employed: pixel-based classification and object-based classification. In pixel-based classification, as seen in figure 2.2, the spectral information of each individual image pixel is viewed as a single input vector in the data set. Since the pixel is the basic spatial unit present in optic sensor data, such as satellite imagery used in remote sensing, this has been the traditional classification approach to machine learning classifiers. However, pixel-based classification is oblivious to the surrounding pixel information, which may hold important details that would otherwise help identify the target pixel's class. Consequently, high spectral heterogeneity classes may show incorrectly labeled pixels within them (referred to as "salt-and-pepper"effect), decreasing the accuracy of the classifier. The inconsistency in data resolution when fusing data from different sources can also lead to a problem of mixed pixels, when multiple pixels from one resolution are present in another. Despite the lack of contextual information provided by the pixel based approach, this approach is still preferred for multi-temporal applications or when the thematic resolution of the land is the important criterion [7]. Most studies compare the efficiency and effectiveness of different machine learning classifiers using pixel-based approaches [46–48], however the multiple factors related with the visual information such as lighting conditions, environmental and spectral variability between members of the same class may negatively impact the classification result when dealing at pixel scale [49].

Other methods such as Object-based image analysis (OBIA) can mitigate some of the problems of the pixel-based approach. [50]. The aforementioned method considers contextual information by incorporating spectral and textural information to identify the thematic class of a pixel cluster within an image. OBIA segments the image into homogeneous objects (i.e. cluster of pixels) based on adjustable parameters as shape, scale and compactness. The classification of the object is subsequently accessed based on statistical properties of the constituent pixels, thus assigning the same class to every pixel, in turn, eliminating the spectral variability present when mapping heterogeneous landscapes with pixel-based methods [51]. Object based approach can also aid in upsampling low resolution maps into suitable sizes. This approach, however, may lead to a decrease in accuracy when applied to irregular classes such as trees and vegetation [52].

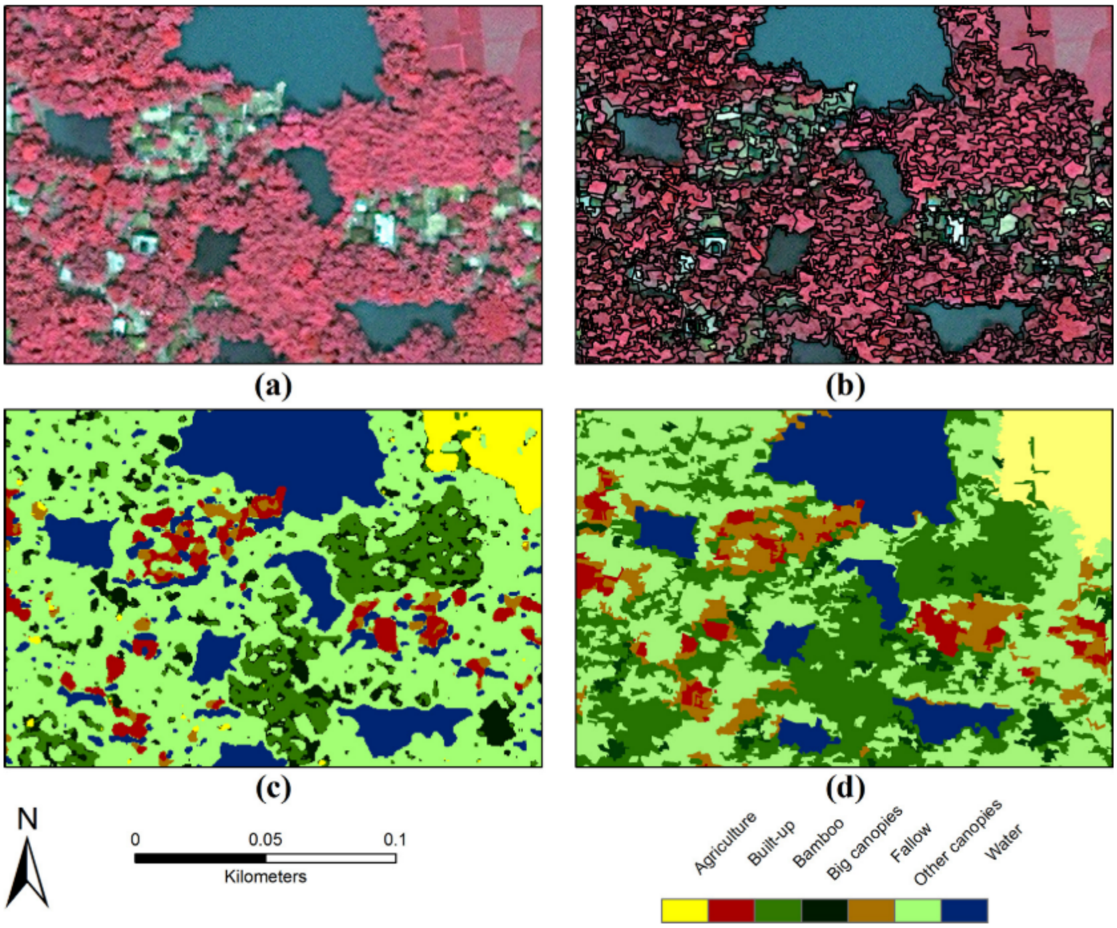


Figure 2.2: Comparison between pixel-based and object-based classification using Random Forest algorithm. (a) represents the pixel-based reference map, (b) represents the object-based reference map, (c) represents the RF classification of the pixel-based approach, (d) represents the RF classification of the object-based approach. In this example, the "salt-and-pepper"effect is clearly visible in the pixel-based approach [47].

Recent published research studies [53, 54] have concluded that object based classifiers show improved performances in overall accuracy between classes over pixel based architectures, especially in studies where high spatial resolution sensors were used [55, 56]. However comparisons of classifiers constructed using multi-sensor remote sensing images are still limited. In [51] researchers trained an object-based RF algorithm using multi-sensor data, revealing gains between 3%-10% in performance over pixel-based RF classification using the same training data indicating the suitability of OBIA for multi-source applications. Texture extraction has also been reported to be aided by object based identification for allowing the identification of relatively homogeneous regions within the image [57].

2.2 Technology introduction

Remote sensing is the most common source of data on land cover and land use applications [7]. Such data includes information assembled by satellites and aerial photography, through the gathering of the reflected radiation produced by the Sun on traditional optic sensors, or self-produced radiation in sensors such as Light Detection And Ranging (LiDAR). Inferences are then made about the earth surface based on the reflectance properties of the surface materials. The use of automated models in remote sensing allows to quickly process image data granting high temporal resolution and replication, crucial capabilities for monitoring applications such as deforestation tracking [58]. In the next sections, a brief introduction to: remote sensing imaging sensors, satellites used in previous studies, the geographic information system used by the plug-in, as well as open-source machine learning algorithms, is presented.

2.2.1 Remote sensing imaging sensors

Remote sensing generally refers to the acquisition or measurement of information of an object or an event, by a recording device that is not present in the near vicinity. Remote sensing in land cover and land use applications is often done from satellites, using a variety of different sensors, with different temporal, spatial, spectral, and radiometric characteristics. Among them, optical sensors, used for over four decades (i.e. Landsat 1 in 1972 [59]), span the range of electromagnetic wavelengths from near, short wave, and thermal infrared to the visible spectrum, assembling images based on the reflected and emitted radiation from the Earth's surface, as illustrated in figure 2.3. The availability and consistency of optical remote sensing has dominated LCLU mapping when compared against field-based methods, specially across large areas [60]. Previously, global mapping programs used mostly coarser spatial resolution (>250 m), however, with advances in sensor technology and the deployment of new satellites, remote sensing implementation can currently achieve high spatial resolution (10 m to 60 m) with free and open data programs [61]. High resolution land cover information has granted the ability to accurately depict land use and to predict atmospheric conditions [62], by reducing the Minimum mapping unit (MMU) (i.e. the minimum size that a land unit must exceed to be able to be represented in the map). It should be noted, however, that higher resolution data is not always the most fitting data, thematic resolution or fit of the land classes may prove to be more important criterion for particular applications [7].

Remote sensing data allows for the monitoring of changes in land cover through time, however continuous analysis of remote sensing data tends to suffer from the changes in classification assessments and inconsistencies derived from improved resolutions of sensors as technology progresses [7]. Therefore, [63] advises for focus on changes in spectral information over time rather than individual interpretations gathered from different years in order to maintain temporal consistency. Spectral data can also be beneficial in class distinguishability in machine learning classification, notably in the delineation of water bodies. Since water absorbs nearly all energy in the near infrared region [64] a simple threshold can often be used to differentiate water from land [65], consequently the delineation of water bodies is often viewed as a straight forward task in remote sensing applications when using spectral data [35]. Researchers have further explored the use of thermal data in classification applications [25, 66], achieving higher identification performances in non-vegetated urban surfaces. Recently [52], have also concluded on the suitability of thermal remote sensing data, showing that Long-wave infrared (LWIR) data is effective as complementary information for land cover/use applications.

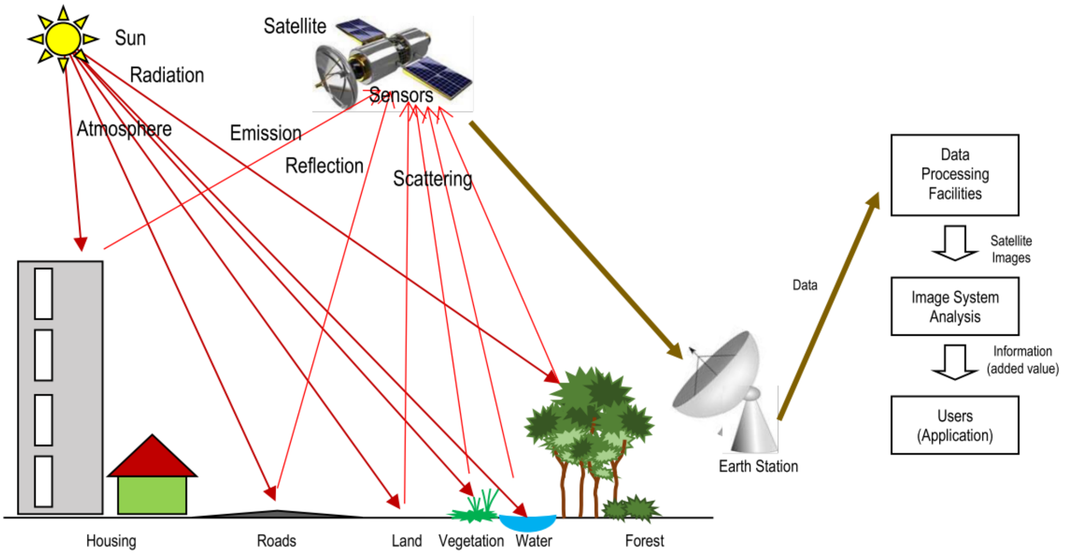


Figure 2.3: Simplified remote sensing process [67].

Multispectral sensors, unlike single channel sensors, survey the land in multiple well defined spectral bands across the electromagnetic spectrum, using filters and instruments that are sensitive to particular wavelengths, this ability allows for the extraction of additional information, based on the spectral reflectance properties of the objects, unnoticeable to the human eye. Different combinations of spectral band can be applied according to the desired observation (i.g. vegetation detection, water depth perception, etc.). Thermal infrared sensors, based on the emitted radiation of the earth, are usually accompanied by poor spatial resolution [68], consequently, they are often reserved for fire detection applications.

While multispectral sensors collect data in a few (usually 3-15) well defined wavelengths, hyperspectral sensors are able to collect exhaustive data across the same range of the electromagnetic spectrum. Such sensors scans hundreds of contiguous relatively narrow spectral bands allowing the assembled data to be analyzed for spectral signatures (i.e. spectral "fingerprints"that identify the materials of the scanned objects). This capability, in turn, grants additional radiometric content over traditional multispectral sensors [69].

In regions where the land presents high spectral and spatial heterogeneity with small well defined communities ,data fusion techniques, such as feature level fusion, which help integrate data and information from different sources may be used in order to increase the classification performance. Particularly when the number of available spectral bands becomes too large to analyze separately. High spatial resolution sensors may also provide data with high inter-class spectral confusion [70], diminishing the suitability of spectral-based methods in very high-resolution spatial images [52]. To overcome this limitation, researchers [71, 72] have resorted to spatial feature extraction methods (i.e. Grey level co-occurrence matrix (GLCM) textures, Object-based strategies, Markov random field, etc.) in order to provide discriminate information.

The aforementioned passive optical data sensors are, however, restricted to daylight acquisition and sensible to atmospheric factors such as clouds [73]. Other techniques such as radar provide its own source of radiation, Synthetic Aperture Radar (SAR) system, for instance, can survey the earth regardless of light conditions [74]. [35] also concluded that SAR data, offers advantages over optical sensor in gathering data during floods and other major hydrological events where clouds are consistently present.

LiDAR is an active sensor primarily consisting of a laser, a receiver and a dedicated GPS. LiDAR uses lasers to supply its own source of illumination, the device transmits a light pulse to the surface, measuring the backscattered and reflected light with sensitive detectors. The distance to the object is then calculated using the difference in time between the transmitted and the backscattered pulses in respect to the speed of light [75]. The resulted data is returned as a point cloud representing the coordinates of the object, which, in contrast with 2D planimetric remote sensing data, portrays a 3D topographic profile of the surface. Two techniques can be used with LiDAR: topographic and bathymetric. Topographic LiDAR is commonly focused on land mapping applications using near-infrared lasers, while bathymetric LiDAR can operate with green light to penetrate water bodies, typically focused on measuring underwater elevations [76]. In land use and land cover applications, LiDAR usually operates in the near-infrared (NIR) wavelength (typically 1064 nm or 1550 nm), since high distinguishability of natural and artificialized land features can be observed [77], allowing the generation of Digital terrain models (DTM) and Digital elevation models (DEM).

Making general terrain classification tools based only on the reflectance characteristics of earth is not ideal, for the same classes may have largely different results following the varied environmental conditions of the different regions of earth [1]. This effect is further verified, since a lot of information is stored in the three-dimensional characteristics of the environment, found to be an important factor in the accuracy of land cover classification [78]. Further studies [1, 79], also concluded that the intrinsic physical attributes such as elevation and slope, are complementary with the extrinsic visual properties, improving the algorithm's performance and complexity with the implementation of geomorphological data. By integrating canopy structural information from different sensors, such as, the previously mentioned SAR system or Light Detection And Ranging (LiDAR), this problem can be minimized [47, 80, 81].

A more detailed characterization of land management can be achieved by fusing data from optical images and radar sensors, allowing for an easier interpretability and delineation of land while including detailed geomorphological information. The inclusion of textural variables in training sets as shown to increase discrimination between classes in land cover and land use applications [82, 83], increasing map accuracy by providing a quantitative description of the image visual characteristics (i.e roughness, smoothness, symmetry, etc),in turn supplying additional information to the classification process [84].

2.2.2 Sentinel-2

Sentinel-2, developed and operated by the European Space Agency (ESA), is an Earth observation mission from the Copernicus programme. The mission is comprised of a constellation of two twin polar-orbiting satellites, Sentinel-2A and Sentinel-2B, launched on 23 June 2015 and 7 March 2017, respectively, phased at 180º to each other in the same sun-synchronous orbit. The mission aims to systematically provide free and open global optical imagery at a high spatial resolution, granting continuity of SPOT and LANDSAT-type image data in monitoring changes in the Earth's environment. The twin satellites carry multispectral sensors at a mean altitude of 786 km, surveying the same location in a 5 day cycle (when employing both satellites). The assigned multispectral sensor provides an orbital swath width of 290 km, monitoring the planet's surface in 13 spectral bands in the near-infrared, short-wave infrared and visible portion of the electromagnetic spectrum. The gathered data ranges from 10 meters to 60 meters in spatial resolution [61], as described in table 2.1.

The European Sentinel-2 mission has provided crucial high-resolution, multispectral data for a broad variety of remote sensing applications, such as rice crop mapping [44], and water bodies detection [85]. In addition to the significant results in land cover and land use mapping, where improvements over older technology are clearly noticeable [86].Nevertheless, since it constitutes a novel technology, a limited number of studies in LCLU mapping using Sentinel-2 data were conducted, consequently, the scope of applicability is yet to be fully explored, thus requiring continued research [32].

Table 2.1: Available Sentinel-2 spectral bands [87].

	Sentinel-2A		Sentinel-2B		–
Bands	Central wavelength (nm)	Bandwidth (nm)	Central wavelength (nm)	Bandwidth (nm)	Spatial resolution (m)
1 – Coastal aerosol	442.7	21	442.3	21	60
2 – Blue	492.4	66	492.1	66	10
3 – Green	559.8	36	559.0	36	10
4 – Red	664.6	31	665.0	31	10
5 – Veg. red edge	704.1	15	703.8	16	20
6 – Veg. red edge	740.5	15	739.1	15	20
7 – Veg. red edge	782.8	20	779.7	20	20
8 – NIR	832.8	106	833.0	106	10
8A – Narrow NIR	864.7	21	864.0	22	20
9 – Water vapour	945.1	20	943.2	21	60
10 – SWIR	1373.5	31	1376.9	30	60
11 – SWIR	1613.7	91	1610.4	94	20
12 – SWIR	2202.4	175	2185.7	185	20

2.2.3 Quantum geographic information system (QGIS)

QGIS (formerly known as Quantum GIS) is an open-source and free geographic information system, that allows for the creation and analysis of geospatial data. The geospatial data can contain multiple layers from different sources to characterize a geographic location, described using a coordinate system. The accessed information consists of two elements, spatial features and attribute data, that can be represented using two methods: vector data, representing geographical objects with geometric elements (e.g. points, lines and polygons), and raster data, used in satellite imagery, comprised of equal-size cells within a matrix. Each cell within the matrix represents a unit area of the surface with attribute information (i.g. elevation, forest cover type, electromagnetic band values, etc.), as exemplified in figure 2.4. QGIS advanced spatial data is available in coverage and geodatabase formats, enabling the use of "spatial reference system"to acquire relationships among the diverse feature classes[88], other formats such as shapefiles, dxf, PostGis and others are also accessible. The system allows for the integration of Python and C++ plug-ins, further extending the QGIS ’s capabilities. QGIS has proven suitability for the implementation of all the machine learning algorithms mentioned in this study [89], with several companies supporting and constantly developing new features.

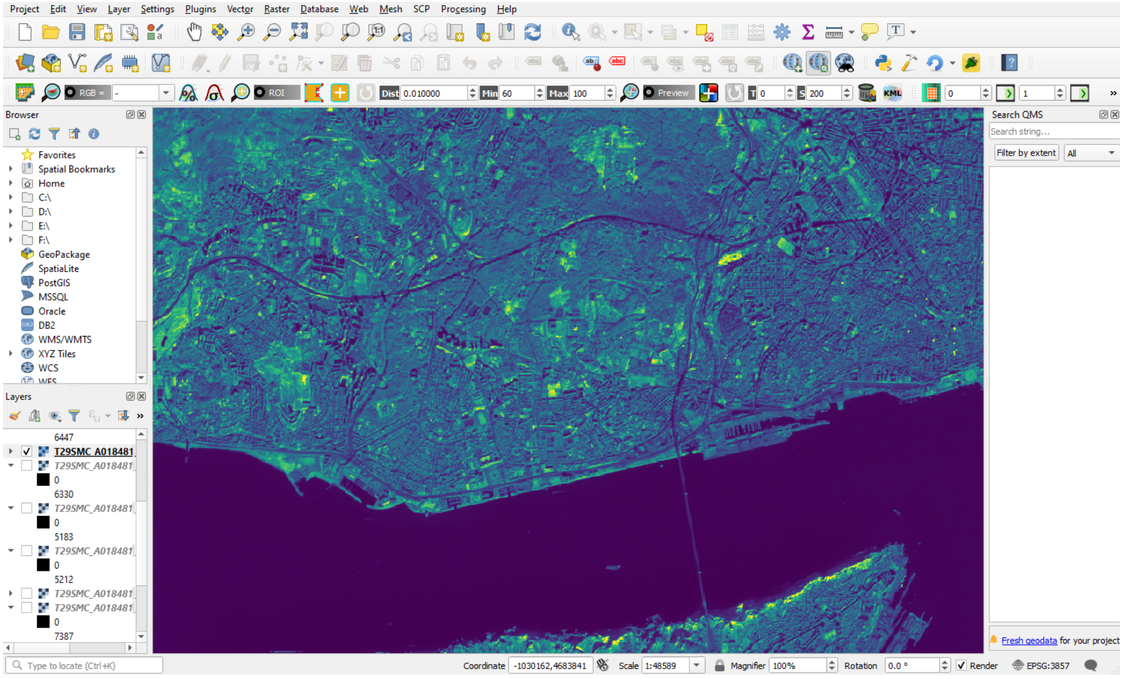


Figure 2.4: QGIS 3.10 software showing Band 8 (NIR) of Sentinel-2 (in pseudo-color) over Lisbon.

2.2.4 Open Source packages

With the increase in complexity and variety of machine learning techniques, it is no longer viable for every research team to manually write all the base code for the algorithms, filled with specified statistical and mathematical formulas for every application. Consequently, open-source code emerges has a mechanism to reduce the inefficient and tedious work surrounding Machine learning tasks, reducing the development time. Over the last two decades, Python programming language, has surged in popularity [90] as a result of the extensive collection of libraries along with the concise, object oriented, and easily readable syntax that it provides to programmers. In addition, since it is also developed under OSI-approved open source license, it's freely usable and commercially distributable [91]. The aforementioned programming language offers numerous frameworks and libraries, particularly in artificial intelligence and machine learning, composed of pre-written code to solve common programming tasks, providing valuable functions with which to build upon. In the scope of this project, the following libraries are pertinent:

- **Pandas :** Offers tools for data extraction, analysis and preparation.[92, 93]
- **Numpy :** Provides functions to deal with complex mathematical operations.[94]
- **Scikit-learn :** Provides various supervised and unsupervised learning algorithms (e.g. Random Forest, Support vector machines), designed to work with Numpy and SciPy scientific libraries.[95, 96]

- **Keras :** Leading open-source Python library focused on providing high-level neural networks API (application programming interface).[97]
- **TensorFlow :** End-to-end machine learning library designed to conduct high-end numerical computations involving tensors, widely used in deep learning research.[98]
- **Matplotlib :** Data visualization focused library, provides 2D plotting and graphs.[99]
- **Theano :** Popular python library, dedicated to the optimization and evaluation of mathematical expressions comprising of multi-dimensional arrays.[100]

2.3 Related work

In the next section a description on the most utilized machine learning algorithms within the reviewed literature is presented. The advantages and sensibilities of such techniques are also displayed, in line with the researchers' conclusions. A final comparison is presented in the last subsection, containing inferences from dedicated machine learning comparison studies.

2.3.1 K-nearest neighbor

K-Nearest Neighbor (KNN) is a non-parametric approach [101] used in statistical applications in the early 1970's [102]. KNN works by mapping a new unseen value onto the mapping of the training data, the algorithm searches the K nearest training samples from the new value and calculates the result with the highest representation. The value K is determinant in the performance of the classifier, in turn, the value should be odd in binary classification to avoid ambiguous results. Increasing the value K leads to a more robust classifier against outliers, but reduces the sensibility to small well defined enclaved regions, for this reason choosing the ideal value of K is not a trivial endeavor [103].

2.3.2 Support vector machine

Support vector machine (SVM) first developed by Vapnik and Chervonenkis 1963 [104] and later modified in the 1990's, is a supervised learning classifier represented by its non-parametric and distribution-free characteristic [105]. Support vector machines separate two different classes by establishing the hyper-plane that achieves the widest margin between the feature spaces of the training sample, this hyper-plane marks the decision boundaries between the two classes[6]. The width of the margin is determined by the distance from the two closest training samples of each class, called support vectors, to each other. In order to classify the probability of a given unseen sample belonging to a certain class, the algorithm measures the distance of the new sample to the training samples taking into account kernel weights, which contain information about the relevance of the input for the discrimination between the two classes. In cases where no linear hyper-plane can be drawn, a nonlinear kernel function is used in order to perform a "kernel trick", this method refers to the transformation of the data map into a higher dimensional feature space, allowing a linear separation to exist and intern describe a originally nonlinear region, as exemplified in figure 2.5b. From the distinct kernels available within the literature, the usage of the Radial basis function (RBF) in Support vector machines has shown to achieve good performances in land cover classification studies [106]. When applying RBF kernel to SVM classifiers, two parameters must be optimized. The width of the kernel is reigned by the value of γ and the cost parameter defined by C [107]. The C parameter adjusts the rigidity of training data, controlling the size of the allowed misclassification in the training data, the value of C is directly responsible for the fitting level of the model [47]. The γ parameter controls the shape and the radios of the hyper-plane, increasing the gamma value leads to a larger and smoother kernel, where a low value generates a steeper and smaller kernel.

Support vector machines, although designed for binary classification can be used in multi-class applications by dividing the problem into a set of binary problems. This method can be achieved with the use of strategies such as one-against-one and one-against-all [69]. SVM classifiers are suitable for solving generalizations problems not linearly separable, the method is robust to noisy and imbalanced remote sensing data [32], furthermore, since SVM calculates the class membership probability of a given sample, it is suitable for soft classification approaches [35]. Additionally, studies have shown that the intrinsic ability of SVM to project hyper-planes from only two support vectors has led to the production of higher map accuracy from small training samples than traditional classifiers [108]. Finding use in a multitude of remote sensing applications, such as rice crop mapping [109].

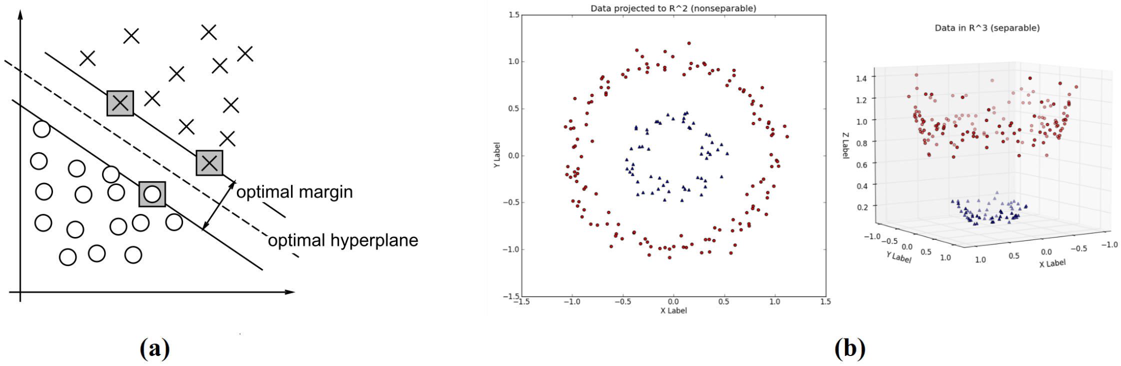


Figure 2.5: Separation of linear and non-linear problems using SVM: (a) linear problem where the support vectors, identified with grey squares, delineate the widest margin [110]; (b) application of a "kernel trick"to a non-linear problem [111].

2.3.3 Random forest

Random forest (RF) classifier, first proposed in 1995 by Ho [112], is an assemble learning method based on Decision tree (DT), the method seeks to avoid the over-fitting tendency and sensibility in changes in the training set that DT manifest [113] by building and training a multitude of decision trees with different split parameters, computing the class with the biggest representation in the output, as seen in figure 2.6. RF classifiers, among the non parametric methods, are becoming increasingly popular in image classification studies in recent years [114] as a consequence of addressing the difficulty with the implementation of large amounts of features available in the training data. The excessive increase of input features may provide additional information to the system, however the increase in complexity derived from the exponential increase of the feature space volume may cause the available data to be insufficient, known as "curse of dimensionality"[115]. By considering a random subset of input variables in the division of nodes, RF reduces the generalization error while allowing the presence of an extensive number of inputs. In order to increase tree diversity, RF uses boot-strapping or bagging to assemble the sub training data sets used on each tree [116]. The assembly is done by randomly selecting samples from the original data set with replacement, meaning, without removing the selected data from the original data set, allowing the next subset to contain every possible combination of inputs [84].

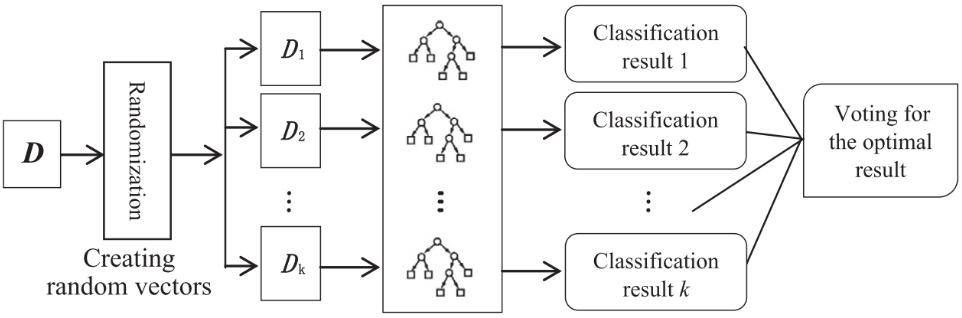


Figure 2.6: Underlying structure of Random forest classification [117].

The implementation of Random forest classifiers requires the optimization of two parameters. The number of trees within the model and the number of features considered in each split [118]. The tuning of these parameters is determined by the specific application, however, studies have shown that the default values of commonly used implementations of the algorithm are satisfactory [119], and while the increase of trees is not detrimental to the performance of the classifier, it can lead to unnecessary complexity [116, 120]. [121] also suggested that over-training of the model is not correlated with the increase in trees. In addition, a restricted number of predictive variables has also shown to improve the model's accuracy by reducing the correlation between trees. Previous researchers [122], have concluded that RF classifiers have the ability to estimate the relevance of variables in the classification of the data samples, while being less prone to noisy and unbalanced data [123]. Furthermore, RF classifiers have shown higher levels of classification accuracy than parametric classifiers (i.e. Maximum likelihood classifiers) in applications where the input data does not present a normal distribution, such as geostatistical textures [84].

2.3.4 Artificial Neural Network

Artificial neural network (ANN) is a non-parametric machine learning technique based on human neural synapses, it's architecture consist of single processing units, called neurons or nodes, arranged in linked layers to form a network, as seen in figure 2.7. Different network architectures can be implemented considering the intended application, such as Recurrent neural networks for speech recognition and Deep neural networks for image recognition. In remote sensing classification, despite the increase in deep neural network research in recent years, Multilayer perceptron (MLP) architecture is still present within the literature. The aforementioned architecture is a class of feedforward ANN consisting of an input layer, at least one hidden layer, and an output layer [124], where every neuron in each layer is connected, along with an assigned weight, to all the neurons in the adjacent layers. The network is trained by tuning the interconnection strengths of each node, creating an internal representation of a defined input pattern in respect to desired outcome [2], this is achieved with the use of backpropagation techniques on supervised training sets in each iteration.

In forward propagation, the input of every subsequent neuron is represented by the weighted sum of the outputs of the neurons in the previous layer, a bias is then added to allow the function to traverse the y feature space axis, enabling the computation of every starting value. To prevent the saturation of the network, the output value of each neuron is then calculated by feeding the input through a non-linear activation function. Following the acquisition of the final network output, the result is compared against the intended target and the error between them measured using a variety of techniques, such as root mean squared error. Subsequently, the error is back propagated through the network using the chain rule of calculus, tuning the weights and biases accordingly to the relative error calculated by the partial derivatives of the error function in order to approximate the classifier to the minimum error. Consequently, the activation function must be differentiable when applying backpropagation.

With each iteration (i.e. forward and backward passes), the network output function error converges to a minimal value, in turn, the network "learns"by determining the most suitable values for the set of weights associated with each neuron that represent the characteristics of the training samples. The number of iterations controls the fitness level of the network, consequently, too many iterations can lead to loss of generalization, potentially harming the performance of the classifier [125]. Neural network classifiers' non-parametric feature allows them to handle data with varied scales and units effectively [126, 127], thus solving some of the problems associated with land use and land cover classification [46].

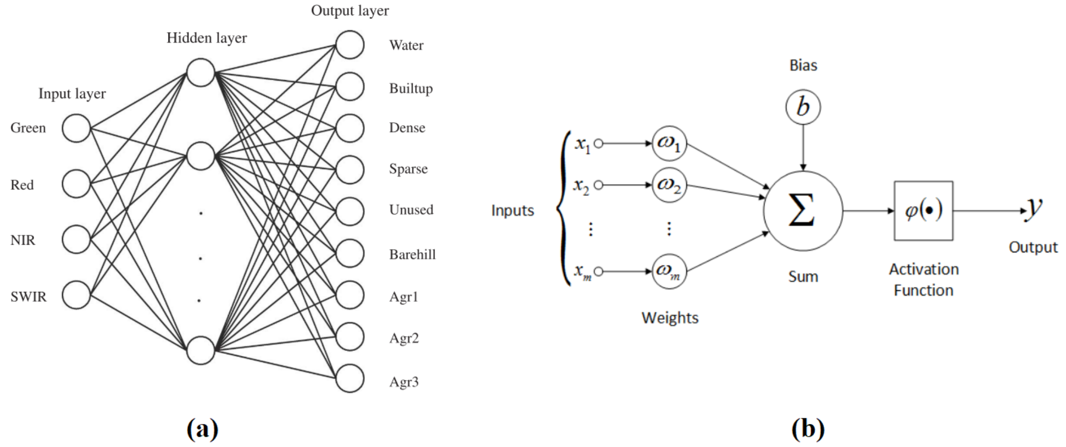


Figure 2.7: Structure of a multilayer perceptron neural network: (a) example a MLP neural network applied to remote sensing [46], (b) representation of the neuron structure within a neural network [128].

2.3.5 Deep learning

Traditional machine-learning techniques, being limited in their ability to process raw data, rely on human intervention in order to interpret features in raw data for further pattern detection. Conversely, representation learning (or feature learning) is a set of techniques that allow a machine to automatically discover and extract features from raw input data. Deep-learning methods are constructed using multiple levels of representation learning techniques. Starting with the raw input, each layer transforms the representation from one level to the next in a more abstract way, progressively extracting higher level features from prior levels. With enough transformations, very complex functions can be learned [129]. Deep neural networks enrich the level of features by stacking layers and increasing the depth of the network, this allows the integration of different levels of features (i.e low level, high level) and classifiers in an end-to-end multi-layer architecture [130]. This machine learning technique has achieved great results, beating previous techniques in a variety of applications, especially on tasks that require pattern reasoning, such as image recognition [131], speech recognition [132, 133], as well as applied physics [134] and medicine[135]. With the increase in depth of the networks, different methods had to be implemented in order to overcome problems such as vanishing gradients and over-fitting. Vanishing/exploding gradients have mostly been addressed by normalized initialization [136, 137] and the presence of intermediate normalization layers [138], allowing the training of networks through the implementation of Stochastic gradient descent (SGD) with back-propagation [139]. To prevent deep neural networks from overfitting, dropout can be used as a regularization technique[140]. Rectified Linear Unit (ReLU) functions were also found to be typical favored over the most common smoother activation functions (i.g sigmoid functions), increasing the learning speed of networks with abundant layers, and removing the need for unsupervised pre-training in deep supervised networks [136]. However, some complications still remain, with the increase in depth of the network, a degradation problem emerges, halting the accuracy at first and rapidly degrading it afterwards. This phenomenon is not attributed to the (over) fitness level of the network as reported by researchers [141, 142]. In recent years, studies [142] explored the use of learning residual functions to mitigate the problem, achieving exceptional results on residual networks with over 100 layers.

Amidst deep neural networks, Convolutional neural networks (CNN) first proposed by Fukushima (1980) [143] are designed to process input and output data in the form of multiple arrays (or matrices) regarded as feature maps, allowing the handling of data in more than two dimensions, as exemplified in figure 2.8. CNN differ from traditional neural networks by four key concepts: use of many layers, local connections, shared weights and polling[23]. CNN architecture is composed of a series of stages, relying on the use of many layers in order to alternate between convolution, non-linearity, pooling and fully connected layers. In contrast with fully connected layers where each neuron receives input from every element in the previous layer (i.e. the receptive field accounts for the every previous neuron), many CNN layers are constructed using local connections, restricting the receptive field to a smaller area. In order to reduce the resources needed to train deep neural nets, in CNN many neurons can share the same filter banks (i.e. weights and biases considered when computing the output function of a neuron) with multiple receptive fields. While convolution layers detect and extract features from the input data, pooling layers merge similar features into one [23], this is achieved by combining the outputs of neuron clusters in a previous layer to one single neuron in the next layer, in turn, reducing the spatial dimensions of network model. The significant implementation of this technique in image analysis applications, has led to an increase in remote sensing use in recent years [129].

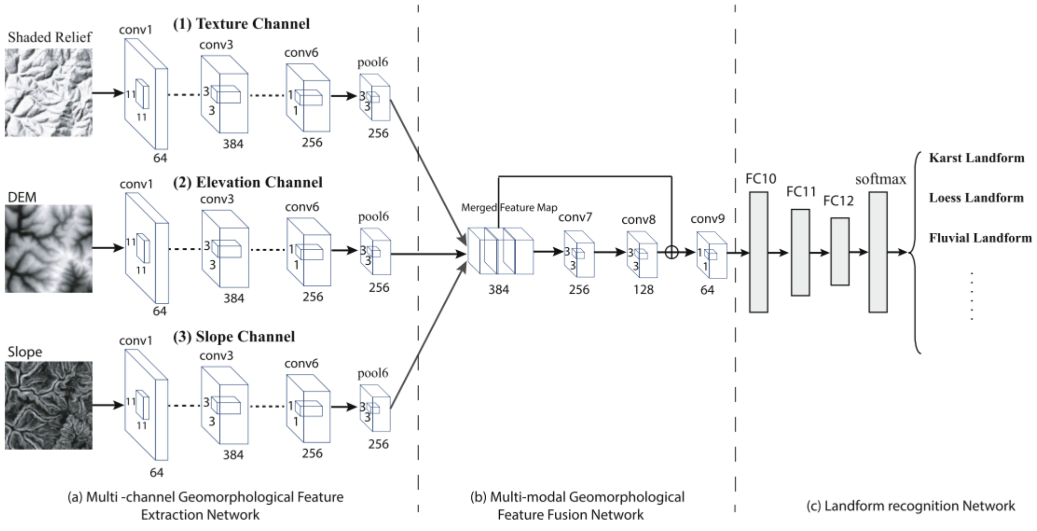


Figure 2.8: Example of an autonomous terrain recognition implementation using Convolutional neural networks within a multi-modal framework [79].

2.3.6 Comparison

Within the literature, it is clear that the choice classifier is bound to the intended application, reigned by factors such as: intended resolution, available data, sensory sources and temporal dependency. Random forest classifiers which are based on machine learning theory [118], and Support vector machines introduced in the past decades to remote sensing [144], have been applied to a variety of applications in recent years, from medical imaging to environmental and land cover monitoring [117, 145, 146]. These classifiers have been able to achieve reliable mapping results [47, 147], while outperforming artificial neural networks, binary hierarchical classifier, decision trees [74, 148, 149]. However, information on the application of deep artificial neural networks in remote sensing is still limited. The use of multispectral data in remote sensing applications has indicated that wavelet transform-based backscatter profiles were able to characterize the seasonal variations within the studied regions, confirming the validity of RF and SVM as suitable machine learning classifiers for mapping rice crops [109].The increase in available hyperspectral data and the high dimensionality that these data sets generate, suggest Random forest as suitable algorithm for hyperspectral image classification [150]. Support vector machines have also shown potential in hyperspectral image classification [6] by exhibiting higher accuracy than traditional classifiers when using smaller data sets, opposing the relative reduction of training data prompt by the increase in dimensionality of the feature space. Other articles such as [32] have also shown, that among SVM, RF and KNN, SVM produced the highest overall accuracy while being the least sensitive to training sample sizes, in line with [151], training data imbalance reflects less with a large enough training sample. Consequently, Random forest classifiers and Support vector machines are generally perceived to achieve the highest overall accuracy results [89].

However, the challenges related with model suitability are still a highly debated topic, with multiple contradictory results from different research teams. As reported by [7], the comparison of studies on land cover and land use classifiers from different researches has been hampered by consistency issues. The studies are often applied to different regions, using different sources of data and evaluated with a variety of accuracy assessment methods. The ability to compare classifiers is further hampered with the use of similar class legends across maps, with different choice of thematic content aggregated to construct the data sets. The extensive existence of machine learning techniques, tuning parameters, and differences in pre-processing , also contribute to the lack of a conclusive resolution [32], with most machine learning routines generating LCLU maps with similar accuracy when given enough data [5].

Chapter 3

Solution strategy

The application of sensory data and machine learning in remote sensing applications is nothing new, however, applications with Portugal as its focus are limited, in addition, the introduction of temporal context to the input data further limits the available academic work. As a result, the following chapter describes the steps taken and the methods used to develop a QGIS focused implementation of an automatic, multi-temporal, land cover and land use classifier. The work-plan will broadly follow the steps introduced in Chapter 2 (i.e. data gathering, algorithm selection, classifier performance test, and accuracy assessment), along with additional concepts and QGIS related steps.

3.1 Central platform

As previously stated, the platform chosen for the LCLU implementation takes the name of QGIS, this platform, among other functions, allows its users to interact with a fixed grid composed of cells, each representing a small portion of the earth surface. Within this surface, the program allows for the creation and management of different types of layers, each with their own coordinate system and data types. The availability of such layers allows for the representation of visual and computational analysis of a variety of data samples(e.g. visual data, topographic data, etc.), in turn, facilitating the recognition and establishment of new and old concepts. Within the platform creative functionalities, the creation of add-on plug-ins is provided, such plug-ins are commonly written in python programming language in order to access the extensive open-source third-party libraries available online. Accordingly, the plug-in built for this thesis proposed implementation will also be constructed on top of the python programming language.

For the purpose of ease of accessibility a basic Graphical user interface (GUI) will also be created, allowing the user to access the plugin functionalities through interactive elements. In order to accomplish this task, an Integrated Development Environment (IDE), namely "Qt Creator", will be used.

3.2 Data gathering

In order to train a machine learning algorithm, data regarding the application goal needs to be acquired and fed into the process during the training procedure. In the present case, in similarity with previous approaches described it the literature, the data required is based on multi-band satellite imagery. In view of the supervised nature of the classifiers being used (described later in the report), this data must be gathered in relevance with the available labeled data. Consequently, this section will be divided into two sub-sections, each describing the steps employed and respective sources from which the data, raw and labeled, was collected and assembled as the training data set.

3.2.1 Labeled data

In the interest of making the project viable for country wide use, a comparable or equally broad labeled data was fundamental. Given the relevance of such data to territory management activities, a national program launched in 1990 by *Direção-geral do Territorio* (DGT), a Portuguese national public organization, had been assembling, compiling and manually updating a nation wide map every few years (i.e. so far, 1995, 2007, 2010, 2015 and 2018). The aforementioned map takes the name of *Carta de Uso e Ocupação do Solo* (COS) [152]. The COS map consists of a polygon mesh, with the minimum area of one hectare each, describing up to 83 different classes, in which the classification is granted based on a percentage of over 75% of representation of any single class within the respective polygon. Given the usefulness of the initiative, the COS map created in 2018 will be chosen as the ground truth for the labeled data used in the training of the project's algorithms.

As shown in previous approaches described in the literature [16, 153], given the overlap of some classes features the difficulty of classification increases in a nonlinear manner with the increase of the number of classes to classify, consequently it is necessary to reduce the number of classes to a manageable value in order to maintain acceptable level of accuracy. Therefore, in line with the number of classes classified in previous approaches, the 83 available class labels in the COS map will be first condensed into its nine major classes, with these major classes representing the core constituents of the thematic map, as illustrated in figure 3.1. In accordance, follows a brief description of each major class and their respective number used in the final symbol labeling:

1. **Artificialized** - The artificialized major class consists of the surface area destined for human interaction. It encompasses, among others: railroads, highways, industrial areas, urban areas, tourism and service areas, airports, quarries, sport and camping facilities, as well as parks and open areas in the vicinity of built up spaces.

2. **Agricultural** - The agricultural major class consists of the surface area employed for agricultural activities. It encompasses, among others: permanent and temporary cultures, vineyards, rice paddies, orchards and olive groves.

3. **Pasture** - The pasture major class consists of the any surface area, cultivated or otherwise, composed of at least 25% grass like vegetation where a system of crop rotation is not employed. It is a fairly restricted class, encompassing mainly pasture with up to 10% of forest trees coverage.

4. **Agroforestry** - The agroforestry major class consists of the surface area where intercropping, between pasture fields or permanent cultures and certain forest species, is present at a minimum of 10% coverage. It encompasses, among others: areas occupied by different species of oak and pine trees.

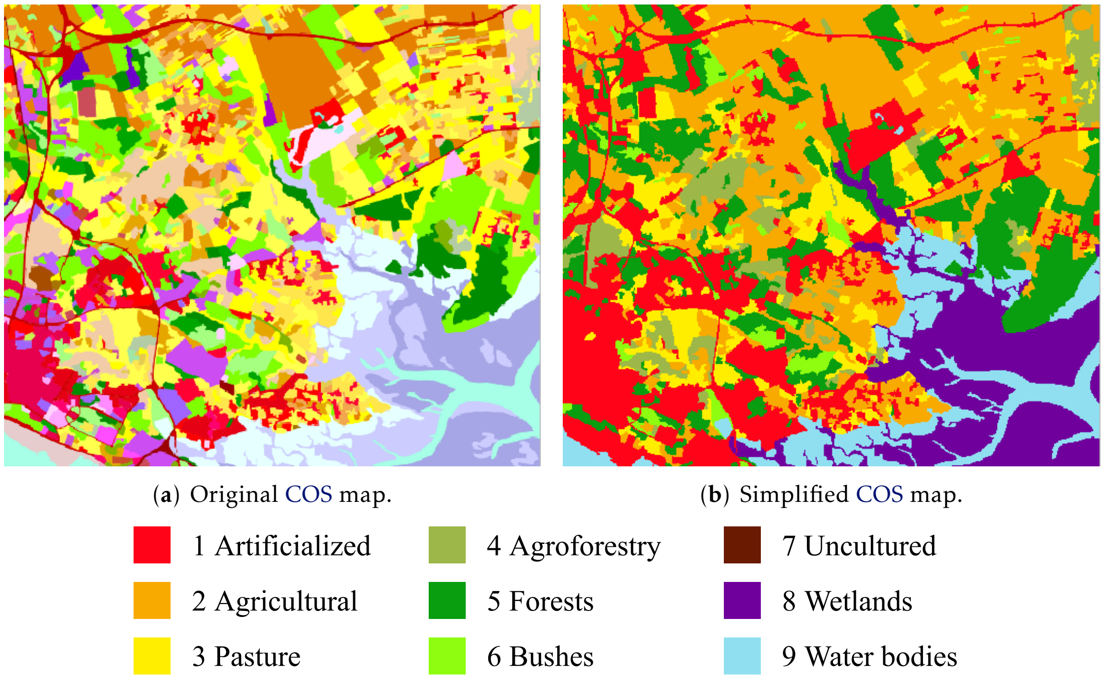
5. **Forest** - The forest major class consists of the surface area reserved to forestal use, either occupied by trees or deforested as result of wildfires or authorized clearcutting, it distinguishes itself from the agroforestry class through the lack of agricultural plots and the minimum height clearance of 5 meters. It encompasses, among others: leafy forests, pine tree forests, different species of oak trees and eucalyptus forest.

6. **Bushes** - The bushes major class consists of the surface area primarily composed of spontaneous vegetation, in which shrub like vegetation accounts for at least 25% of the area. It is a relatively restricted class, encompassing mainly bushes and areas with sparse presence of groves.

7. **Uncultured** - The uncultured major class consists of any natural surface area in which vegetation is scarce or nonexistent, in rigorous terms, describing areas with less than 25% coverage of bushes or grass. It encompasses, among others: beaches, rock formations and uncovered land.

8. **Wetlands** - The wetlands major class consists of the surface area describing soils with considerable prevalence of water, either annually or through rotational cycles. It equally accounts for salt and fresh water, encompassing, among others: wet ligneous zones, coastline regions, marshland, as well as intertidal zones.

9. **Water bodies** - The water bodies major class consists of the surface area covering large natural or artificial bodies of water. It describes both transitioning and still water surfaces, encompassing, among others: water courses with more than 20 meters of minimum width, water planes, lagoons, dam reservoirs, lakes, salt pans, rivers, seas and oceans.



(a) Original COS map.

(b) Simplified COS map.

(c) Class labels.

Figure 3.1: Simplification of COS map.

3.2.2 Raw pixel data

With a secure source for the ground truth information collected, follows the acquisition of the raw ground pixel data from the satellite imagery. As previously indicated in chapter 1, and later clarified in chapter 2, the raw data acquisition process will be conducted through the use of the relatively new orbital imagery database produced by the Copernicus programme twin-satellites, sentinel-2A and sentinel-2B. Within the aforementioned database, two sets of products are particularly relevant for the project, level 1-C, providing top of atmosphere reflectance, and level 2-A, providing atmospherically corrected bottom of the atmosphere reflectance. Generally, level 2-A products provide greater band contrast and less weather sensitivity, in exchange of 1 multi-spectral band, specifically the short wave infrared band number 10, described in table 2.1. Nevertheless, for the purpose of this project it is expected that the algorithms ought to benefit more from the higher contrast provided by the Level 2-A products, in spite of one less SWIR band, consequently these products will be chosen as the source of the raw pixel information.

Following the choice of the acquisition source, the data set creation process starts with the assignment of the study sites. In order to reduce unexpected conditions during the normal classification procedure, the training data set must account for thematically diverse places, consequently, locations from north, center and south of the country must be gathered to cover the different elements species within each class. In addition, in order to preserve the accuracy consistency throughout the year, the training data set should be made of temporal data from each of the four seasons.

Within the scope of the project, and possibly the distinguish factor, prevails an analysis and comparison of the training outcome between a single time frame approach(i.e. with only the most up to date imagery), and a multi temporal approach(i.e. using a sequence of images, from the recent past, up to the present in a coherent form). Consequently, the training data set should consist of data gathered in groups of multiple seasons each in a sequential order. For the pursued solution three intervals were considered. As a result, the training data set requires the span of six seasons, in reference to the yearly four seasons plus the additional two seasons necessary for the oldest group.

Admittedly, the aforementioned approach is bound to introduce some errors associated with the temporal variance between the data acquisition and the labeling composition date, degrading the training accuracy. This complication, however, is intrinsic to most wide classification procedures, resulting from the temporal and technical discrepancies between the sources, reported, likewise, in the manually-made COS map [152].

At the time of writing, the Winter season for 2018, representing the period from 21 December 2017 through 20 March 2018 was not available in the Copernicus Long Term Archive (LTA), therefore the decision to push the initial date to Spring 2018 was made, in an effort to best achieve a balance between the past and the future relative to the labeling date. Consequently, considering the creation time span of the COS map (i.e. updated in the later half of the year, from June 2018 through October 2018 [152]), the training data set shall include multi spectral data from March 2018 through September 2019, as displayed in figure 3.2.

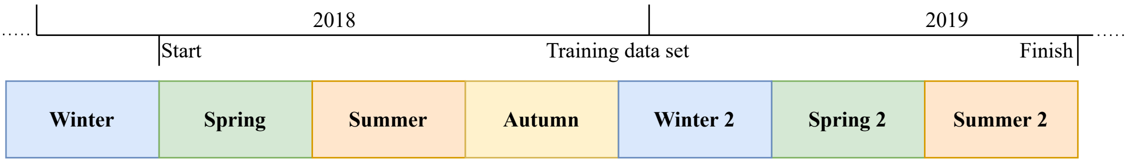


Figure 3.2: Temporal interval of the training data set.

The creation of the seasonal groups follows a decreasing date order, from the most recent to the oldest acquisition. The creation of these groups will then allow for the comparison between the single time frame approach, through the use of the first column bands, and the multi temporal approach, through the use of all three column bands. The groups can be seen in table 3.1.

Table 3.1: Group seasonal construction.

Group	Season progression columns
1.	Summer2 - Spring2 - Winter2
2.	Spring2 - Winter2 - Autumn
3.	Winter2 - Autumn - Summer
4.	Autumn - Summer - Spring

3.3 Data pre-processing

Following the acquisition of reliable data sources, the next step in machine learning procedures is data pre-processing. The benefits of adjustments and modifications to the available data carry drastic improvements to the overall outcome of machine learning algorithms, as such, it is concluded that the gathered data needs to be pre-processed in order to produce a suitable data set [5]. Therefore, in the coming subsections, techniques related with: eccentric and missing data correction, matching of spatial characteristics between the distinct sources, class balance, randomization and data set management, will be described. Additionally, the introduction of vegetation indices to the data set will be presented, together with their computation and areas of focus.

3.3.1 Resolution matching

Given the different native resolutions of the various sensors present in the twin satellites, sentinel-2A and sentinel-2B, the gathered data is not spatially coherent under all of the available bands, ranging from 10 meters per pixel to 60 meters per pixel, as visible in table 2.1. As a consequence, the collected data needs to be scaled. At this stage, two options can be considered, upscaling and downscaling. The upscaling of all the data to 60 meters per pixel based on the average value of the pixels previously situated in the area will result in the creation of new pixel band values, some of which might not represent any class, with existence solely in the numerical domain, resulting in the degradation of the accuracy of the classifiers along with a loss of image granularity. Conversely, the downscaling of all the data to 10 meters per pixel based on the assumption that the new pixels generated from the division of the larger pixels hold the same value, won't create values of mathematical nature, whilst taking advantage of all the available absolute pixel information. Consequently, the described downscaling approach will be chosen in order to match the highest resolution provided across the available bands. The aforementioned technique will be implemented through the "nearest neighbor"method, as illustrated in figure 3.3, accordingly with the prevention of calculated pixel values.

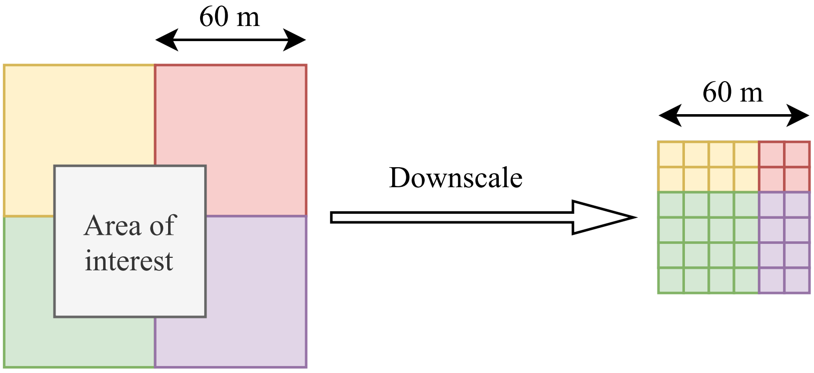


Figure 3.3: Nearest neighbor downscale method.

Similarly, acquired as a polygon layer with infinite resolution, the provided labeled COS map, facilitated by the native dimensional format, will be converted to a raster layer with cell width of 10 meters in accordance with the resolution of the processed multi-spectral bands. Ultimately creating a spatially coherent data set.

3.3.2 Index creation

In order to improve the overall accuracy of machine learning algorithms that lack the capability to correlate the division of input information (e.g. Neural Networks), additional index data, based on calculations previously described in the literature, ought to be used as input in the training and classification procedures.

Vegetation indices can be described as spectral transformations between multiple electromagnetic wave bands that enhance the different land properties based on the diverse reflectance characteristics of its compounding elements (i.e. different species of vegetation, rock formations, water bodies, among others). The aforementioned vegetation indices, although copious in number and limited to its intended application, have shown results in land cover classifications, with particular interest in high spatial resolution satellite imagery [154], where continuous time series allow for comparable and consistent data. Consequently, seven vegetation indices will be used as input data for each season across every data set group, the vegetation indices chosen will target areas prone to overlap, easing the classification between the respective classes and reducing the corresponding errors.

Follows a brief description of every vegetation index used, the targeted area and respective equation:

1. **Normalized difference vegetation index (NDVI)** is a widely used index focused on the reflective properties of the chlorophyll pigment present in plants, the index utilizes the red and near-infrared bands, band 4 and band 8 in the sentinel-2 data set, respectively, to enhance the relative biomass of the area. The selection of the NDVI is targeted at improving the differentiation between the *Bushes* and *Forest* major classes.

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{3.1}$$

2. **Modified Normalized Difference Water Index (MNDWI)**, based on the reflective properties of water, utilizes the green and short wave infra-red bands of the spectrum, band 3 and band 11 in the sentinel-2 data set, respectively, to strengthen the characteristics of ample water bodies. Although the classification of water bodies apart from land is often considered to be of trivial nature [65], the distinction of open water bodies from swamp like areas represents an additional challenge. Therefore, the selection of the MNDWI is targeted at improving the distinction between the *Water bodies* and *Wetlands* major classes.

$$MNDWI = \frac{Green - SWIR}{Green + SWIR} \tag{3.2}$$

3. In consideration of the two short wave infra-red bands provided in the Level 2-A products gathered from the sentinel-2 database, an alternative version of the MNDWI previously described, will be used as the third index. The referred implementation utilizes the green and the second SWIR bands, band 3 and band 12 of the sentinel-2 data set, respectively. Similarly, this implementation aims to improve the differentiation of water-filled areas.

4. **Normalized Difference Moisture Index (NDMI)**, attempts to estimate the moisture levels present in both soil and vegetation, the index employs the the NIR and SWIR bands, band 8 and band 11 of the sentinel-2 data set, respectively, in order to describe the area's water stress level. The selection of the NDMI is targeted improving the categorization of the *Agricultural*, *Pasture* and *Uncultured* major classes, each containing zones of overlap.

$$NDMI = \frac{NIR - SWIR}{NIR + SWIR} \tag{3.3}$$

5. **Normalized Difference Built-up Index (NDBI)**, supported by the higher SWIR reflectively observed in man-made structures [155], utilizes the NIR and SWIR bands, band 8 and band 11 in the sentinel-2 data set, respectively, to emphasize built-up areas. The selection of the NDBI is targeted at increasing the confidence level of the *Artificialized* major class, in turn, reducing the tendency, occasionally present in the manually made labeled map, of ignoring roads.

$$NDBI = \frac{SWIR - NIR}{SWIR + NIR} \tag{3.4}$$

6. **Bare Soil Index (BSI)**, carrying the ability to highlight non-agricultural from agricultural areas [156], utilizes the Blue, Red, NIR and SWIR bands, band 2, 4, 8 and 11 in the sentinel-2 data set, respectively, to discern bare soil from farming land. The selection of BSI is targeted at strengthening the difference between *Agricultural*, *Bushes* and *Uncultured* major classes.

$$BSI = \frac{(Red + SWIR) - (NIR + Blue)}{(Red + SWIR) + (NIR + Blue)} \tag{3.5}$$

7. **Normalized Difference Greenness Index (NDGI)**, based on the reflective properties of different pigments, utilizes the Green and Red bands, band 3 and 4 in the sentinel-2 data set, respectively, to evaluate the variation and quantity of various pigments throughout the vegetation season [157], allowing for better distinction of flora-based classes in continuous time series. The selection of NDGI is targeted at improving the categorization of the *Agricultural* and *Pasture* major classes.

$$NDGI = \frac{Green - Red}{Green + Red} \tag{3.6}$$

3.3.3 Learning quirks

In addition to the aforementioned required functional steps (e.g. resolution matching), some additional data manipulation can be done in order to increase machine learning algorithm's effectiveness, particularly in backpropagation approaches [139]. In the present subsection follows a brief introduction of the prospective methods, provided as suitable techniques as per the available literature.

The first technique can be described as "class balancing", this technique can be applied to prevent the under or over representation of some class in the training data set. The aforementioned method allows for a more balanced classification procedure when applied in real world applications, in turn, reducing bias and derived errors. Considering the high variance of abundance of each major class within Portugal, the technique manifests itself in the acquisition of multiple small determined zones in contrast with fewer large zones, thereby providing a bigger control over the tuning of each class samples.

The second prospective technique consists in "data shuffling", in essence, the method describes the creation of a supplementary data set based on the random selection, without replacement, of data points from the original data set. The creation of a statistically consistent data set allows for the reduction of bias development in the early stages of the training procedure. In addition, it reduces chance-related precision variance observable in distinct data assortments, therefore diminishing the need for cross-validation in large data sets. Given the sequential nature of the data gathering process in the present project, whereby every data point, or pixel value, is gathered from the vicinity of the previous pixel, the employment of the data shuffling method is of utmost importance in order to prevent long sections of identical data points.

The third technique, normalization, is related with algorithm convergence. Normalization within the scope on the project data is accomplished through the use of two methods, centering and scaling. Shifting and centering the data around the value zero has direct impact in certain ML algorithms' learning rate, in particular, where progressive updates are made based on an error function (e.g. backpropagation neural networks) [139]. Scaling the data to nearly identical covariance also has impact in the algorithms convergence, however, with careful adjustment it can further address outlier related inconveniences. Consequently, normalization is likely to reduce the effect of eccentric values during the classification procedure, in turn, reducing the "salt and pepper"effect occasionally found in satellite image based recognition.

The fourth and final technique is associated with modular classification. In view of the per-class classification architecture, the existing nine major labels ought to be decomposed into nine binary problems, consequently, "one-hot encoding"method shall be attached to the data set with the values -1, 1.

In conclusion of all the data pre-processing and manipulation, the final data set structure is complete as seen in figure 3.4. Each data row of the data set relates to a single unit of the map and is comprised of the three required seasons for the multi-temporal approach, each containing the 12 available spectral bands and the 7 derived vegetation indices. Subsequently, the labeled class and the specified encoding value is added for training purposes.

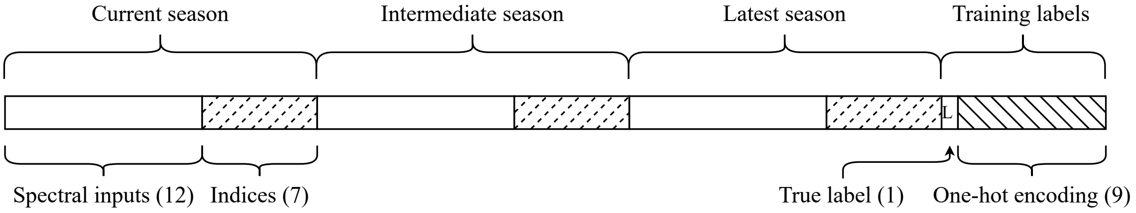


Figure 3.4: Training data set structure.

3.3.4 Data set management

Following the completion of the data set, constructed with shuffled data from the four groups described in table 3.1 and respective indices, succeeds the division of the data set into three categories. A training subset comprised of the majority of the data gathered, used in the learning procedure of the algorithms. A cross validation subset composed of separate data, used every iteration for comparison and optimization of varied algorithms' hyperparameters. And a test subset created from unique, out-of-sight data, used solely in the final accuracy assessment in order to achieve an unbiased result, comparable with real world performance.

3.4 Classifier construction

The next step in the plugin development corresponds to the architecture and framework selection. Being the core of the implementation, this stage is of utmost importance to the overall success of the application. For this reason, some restrictions have to be considered from the early stages of development, such as software requirements and resources feasibility. Consequently, while the extensive training procedure will be presented in chapter 4, the choice of architecture along with the framework selection will be described in the following subsections.

3.4.1 Classifier architecture

In light of the dynamic nature of the specifications of major class labels, a modular approach was chosen in order to create some degree of scalability. The input data shall be fed into nine binary classifiers, each representing a degree of confidence for each of the nine major classes. The aforementioned classifiers outputs shall then be fed into a multi-class classifier in order to obtain the final predicted class, as seen in figure 3.5. This approach allows for an easier integration of additional classes to the classification procedure, as well as decomposition of previous classes, requiring solely the retraining of the altered binary classifiers along with the the final stage classifier. Considering the higher dimensions of the feature space associated with the initial fifty-seven inputs of the first stage in contrast to the nine inputs of the second, the selected architecture composes algorithms of lower orders of complexity, in turn, allowing for faster training and hyper parameter testing.

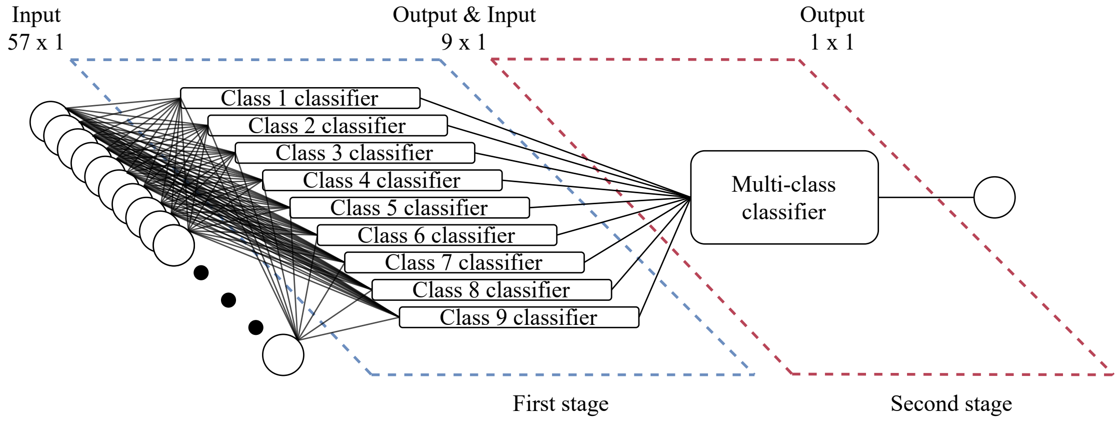


Figure 3.5: Classifier architecture overview.

3.4.2 Framework selection

Given the implementation overhead implications and auxiliary software package requirements, simplicity should be the driven principle when choosing the tools required to operate the plugin. Consequently, despite the high supply of open-source python packages, the use of packages from outside the original QGIS repository (i.e. OSGeo4W) should be kept to a minimum. Considering the previously described software packages in section 2.2.4, problems related with data and feature extraction should be handled by *Pandas* and *Gdal*, a translator library for raster and vector layers focused on geospatial data formats. *Numpy* will facilitate data manipulation and required mathematical computations. Although the aforementioned packages exist pre-installed in the OSGeo4W environment, the basic repository lacks machine learning related libraries, for this reason, it is necessary to choose a suitable external framework. Thus, two options were considered, *Scikit-learn* and *Tensorflow* plus *Keras*. The first option, *Scikit-learn*, describes a high-level library offering off-the-shelf machine learning algorithms such as support vector machines, Random Forests, and Regression Neural Nets, it is the less complex and lighter option. The second option, *Tensorflow*, describes a lower-level library, comprised of comprehensive tools focused on the creation of ML algorithms from the foundation up. *Tensorflow* is particularly appealing for deep learning implementation, due to the advantageous GPU integration available throughout the training procedure. Despite the integration, in recent years, of *Keras* API into the *Tensorflow* framework, providing higher-level, more accessible, ML creation tools comparable with *Scikit-learn*, the incorporation process is still relatively complex in comparison with the installation process of *Scikit-learn*. In addition, considering the per-pixel classification nature of the project, a deep learning approach, traditionally focused on object classification, is not included in initial work plan. In view of the intended implementation, the classification procedure of the plug-in will be constructed as a stand-alone module, as such, *Scikit-learn* shall be chosen for a first implementation for its simplicity and smaller memory requirements. If necessary in future work, the algorithms' framework can be changed. Lastly, the *Matplotlib* package, although not included in the plug-in, shall be used during training for hyperparameter test and accuracy assessment, this library will be responsible for the creation of the graphs displayed in chapter 4.

Succeeding the choice of framework, follows the selection of the machine learning algorithms to be used in the classification procedure. In accordance with the methods reviewed in chapter 2, three ML techniques were chosen in consideration of their key characteristics.

1. **Support Vector Machines**, based on the division of the feature space through the use of an hyper-plane, will be tested as a contender for the first stage of classification in consideration of the binary nature of the categorization. The algorithm, as reviewed in the literature, has demonstrated resiliency to noisy and unbalanced data, such characteristic, despite the attempts to reduce the impact of eccentric data present in the data set (i.e. through pre-processing), is of particular interest given the notably diverse data being fed into the training procedure.

2. **Random Forest**, based on the representation characteristics of smaller diverse decision trees, will be tested as a contender for the first and second stage of classification in consideration of the large initial feature space derived from the large number of inputs. Through the selection of subsets of inputs for each decision tree, the algorithm has been reported to address and reduce the "curse of dimensionality", in turn, diminishing the conceivable detrimental effects of the increase in input count as a result of the included auxiliary index values. Furthermore, throughout the studied literature, random forest algorithms have been increasingly present as a solution for land use and land cover implementations.

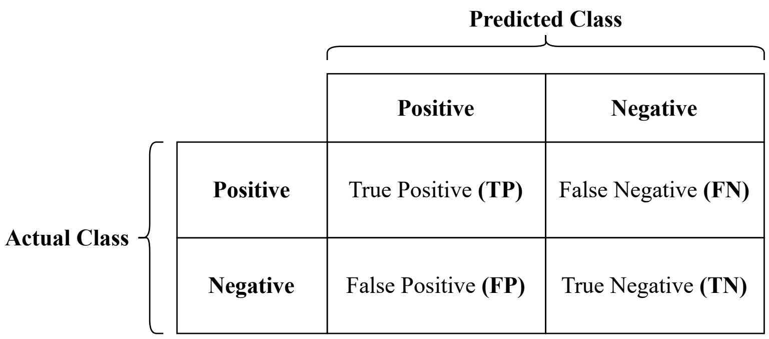
3. **Feed forward neural networks**, based on consecutive layers of interlinked neurons, will be tested as contenders for the first and second stage of classification in consideration of the multi-temporal aspect of the sought approach. In contrast with the other previously mentioned algorithms, the capability of neural networks to employ information from every input at every step of the classification procedure is expected to utilize the contextual information present in the continuous temporal data more efficiently, in turn, achieving higher classification performances. In order

to test the most suitable interface between the first stage outputs and second stage inputs, two architectures will be tested. Firstly, a multi-layer perceptron classifier will be employed has form of an hard classification interface, the algorithm will feed the second stage with a well defined binary value. Secondly, a multi-layer perceptron regressor will be utilized has form of a soft classification interface, the algorithm shall feed the second stage with a value representative of the probability of the output class. In addition to the expected greater usage of contextual information, neural networks, as per the literature, have also demonstrated the ability to handle data with varied scales and units effectively [126, 127]. Such characteristic is crucial given the different nature of the inputs (i.e. integer values from spectral band, and ratios from the calculated indices).

3.5 Accuracy assessment

Within the universe of data analysis and algorithm optimization, many metrics exist with the purpose of accessing a solution's performance. These metrics aim to complement the traditional accuracy evaluation by emphasizing different aspects of the achieved outcome, aspects often sensible to the structure of the used data and concealed in a universal method. For the purpose of the project, these methods are of significant importance considering the modular architecture and resulting binary sub-classification of the intended multi-class implementation. In which the data is roughly divided in a 1:8 ratio of positive instances (i.e. the classifier evaluates its own class) and negative instances (i.e. the classifier evaluates every other class), resulting in a highly unbalanced subset where any classifier that continuously checked every data point as negative would wield a traditional overall accuracy close to 89% despite not doing any classification. Consequently, in this section, supplementary metrics of accuracy assessment will be described along with the implemented formulas.

The base structure, from which all the other metrics are derived, is known as confusion matrix, described in figure 3.6. The structure presents a contingency table displaying the outcome of the classifier through the use of rows and columns containing the true and the predicted values, respectively.



		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive **(TP)**	False Negative **(FN)**
	Negative	False Positive **(FP)**	True Negative **(TN)**

Figure 3.6: Confusion matrix.

Consequently, four groups are established, True positive (TP) representing the correctly identified positive instances, True negative (TN) representing the correctly identified negative instances, False positive (FP) representing the incorrectly identified negative instances, and False negative (FN) representing the incorrectly identified positive instances. Constructed from the aforementioned groups, four metrics will be used in order to access the algorithms' performances.

1. **Accuracy score**, mainly considered in the second stage of classification, computes the traditional ratio between the correct identified values, both positive and negative, and the total number of samples. The range extends over [0, 1].

$$Accuracy\ score = \frac{TP+TN}{TP+FP+FN+TN} \tag{3.7}$$

2. **Recall score or True positive rate (TPR)**, computes the ratio between the number of correctly identified positive instances, and all the actual positive instances, in essence, describing the proportion of positive identified values among all the positive values present in the data set. The range extends over [0, 1].

$$Recall\ score = \frac{TP}{TP+FN} \tag{3.8}$$

3. **Precision score or Positive predictive value (PPV)**, computes the ratio between the number of correctly identified positive instances, and all the positive classified values, in essence, describing the proportion of true positive values among all the predicted positive values. The range extends over [0, 1].

$$Precision\ score = \frac{TP}{TP+FP} \tag{3.9}$$

4. **F1 - score**, computes the harmonic mean of precision and recall, in essence, combining the properties of precision and recall to derive a metric capable of representing the robustness and exactness of a particular classifier irrespectively of data imbalance. The range extends over]0, 1].

$$F1-score = 2 * \frac{precision \cdot recall}{precision + recall} \tag{3.10}$$

Accordingly, F1 - score should be the deciding factor when choosing between algorithms and hyperparameters during the first stage classification testing. For regression methods in which these techniques cannot be applied without the introduction of a threshold, an error or loss function will be used to analyse the algorithms performance.

3.6 Strategy overview

With the major steps of the proposed work-plan outlined, the strategy is expected to produce a sufficiently robust implementation, capable of addressing problems related with the integration and manipulation of data from distinct software systems. Additionally, the strategy aims to improve the overall application performance by encompassing the employment of a variety of optimization techniques.

In summary, follows the depiction of the strategy diagram, as seen in figure 3.7.

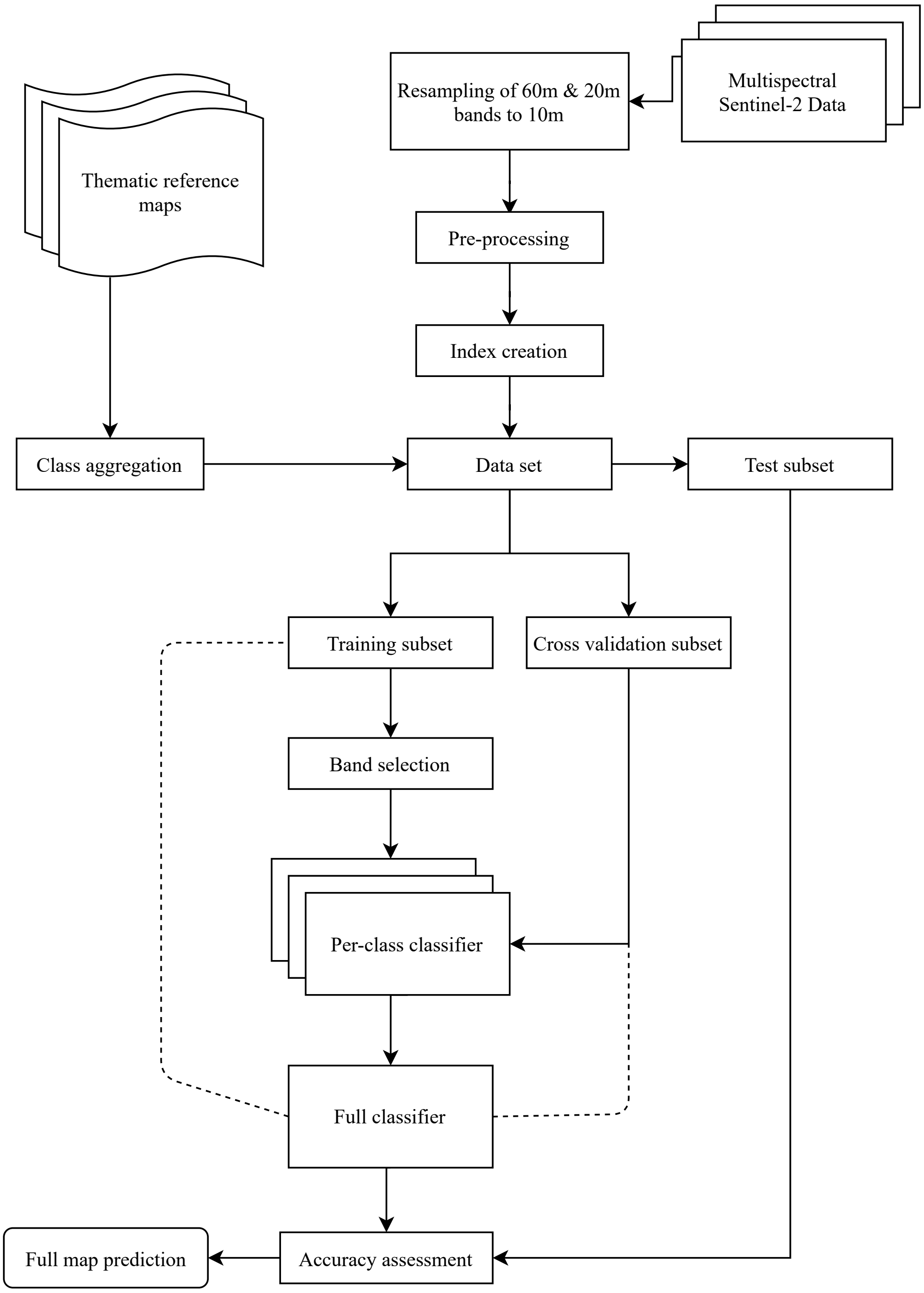


Figure 3.7: Strategy diagram.

Chapter 4

Implementation & findings

Given the interconnected relationship between the implementation decisions and the training results, the present chapter will supplement the steps described in chapter 3, coarsely following and reporting, through logical order, the development processes along with the achieved findings.

Although a substantial amount of work was done to create the QGIS plugin from the ground up, work related with all the connections between the plug-in and the QGIS interface. The ability to draw polygons and extract geographic information from the work canvas, transformations of coordinate systems, extraction of information from the Copernicus database, manipulation of data layers and user interface objects, among others, will not be described as code review is not the focus of the report. Further information regarding the plug-in can be found in Appendix 1.

4.1 Initial data set creation

With the base plugin created and capable of handling, selecting and extracting meaningful data, the implementation starts with the creation of the data set used for the training procedure. As previously designated in chapter 3, COS2018 map was used as the source for all the employed labeled data. The COS map, initially retrieved as a polygon layer containing 83 different classes, was condensed into a nine major class raster layer spanning the whole Portugal's continental land area. The raster layer now described a 10 meter resolution grid containing all of Portugal's land cover classification.

The next step in the creation of the data set, relied on the acquisition of multispectral data from defined regions within Portugal. Through the use of Copernicus sentinel-2 long term archives, 18 image set tiles containing bottom of atmosphere level-2A data were retrieved from three distinct zones, six sets from the northmost portion of the country, six from the center, and six from the southmost section. Each set was gathered in chronological order and together represented the six season group depicted in figure 3.2.

The conclusive step regarding concrete data acquisition comes in the form of the creation of the four seasonal groups described in table 3.1 and ensuing assembly. Given the aforementioned resolution disparity between each spectral band, the bands underwent a process of spatial matching throughout each acquisition action, in which, all bands' tiles were trimmed, downscaled, and aligned with the previously created 10 meter labeled raster layer. In total, 40 areas were selected from across the three major chosen zones, totalling 160 acquisition actions. Each area was chosen based on class representation and with the aim of including, to the limits of feasibility, the biggest range of diversity described by each major class. During the data acquisition process, some adjustments had to be made to prevent extreme class unbalance. The selection process for the sub-zones was adjusted in order to better approximate each class representation to 8% of the total population, with the exception of the reasonable bigger classes (i.g. *Agricultural* and *Forest*). These adjustments, despite leading to the unrealistic representation of some classes (i.g. *Wetlands* major class finds close representation to the *Artificialized* major class, which is clearly not conceivable in a country such as Portugal), were necessary to prevent the classifier from neglecting under-represented classes.

With the data collected from the 160 acquisition actions, the first rudimentary data set was compiled, however, in order create a statistically consistent data set, a new shuffled data set was created through the random selection without replacement of the data points from the original rudimentary data set. The newly created data set containing 6.425.998 data points was then tested for statistical consistency through the creation of two subsets, data set A containing the first 450.000 data points and data set B containing the last 300.000 data points of the complete set, the results can be seen in table 4.1.

Table 4.1: Class representation in each data set.

–	Complete set		Subset A		Subset B	
Major classes	Number of data points	%	Number of data points	%	Number of data points	%
Artificialized	487.073	7,58	34.017	7,56	22.546	7,52
Agricultural	1.453.305	22,62	101.754	22,61	68.141	22,71
Pasture	584.412	9,09	40.789	9,06	27.500	9,17
Agroforestry	619.831	9,65	43.430	9,65	28.530	9,51
Forest	1.307.660	20,35	91.548	20,34	61.127	20,38
Bushes	795.077	12,37	55.814	12,40	37.015	12,34
Uncultured	357.926	5,57	25.025	5,56	16.838	5,61
Wetlands	289.279	4,50	20.242	4,50	13.508	4,50
Water bodies	531.435	8,27	37.381	8,31	24.795	8,27
Total	6.425.998	100,00	450.000	100,00	300.000	100,00

Through the comparison of the classes' representation between the complete set and the derived subsets, as seen in table 4.1, it is possible to calculate an average deviation and max deviation of 0,017% and 0,04% for subset A, and 0,05% and 0,14% for subset B, respectively. In turn, validating the applied shuffle method as a suitable technique to reduce subset related accuracy discrepancies.

Following the strategy plan, the next step in the creation of the data set would be the normalization of the data point values, however, given the active nature of the scaler (i.e. being a functional and configurable element of the training process), its implementation shall be described in greater detail in the next subsection along with the algorithms' training procedures. Consequently, the creation of the data set concludes with the computation and addition of the vegetation indices along with the one-hot encoding data to the complete data set, conversely recreating the structure visible in figure 3.4.

Lastly, in order to reserve data for the final out-of-sight accuracy assessment, the data set is divided into one training data set containing 70% of the data points and one test data set containing the remaining 30% of the data points. Given the statistical consistency observed in table 4.1, the training data set will be further divided into smaller sets during the training procedure in order to facilitate testing and tuning of the different hyperparameters.

4.2 Training procedure

In this section the steps taken throughout the training procedure will be described along with the achieved results. The procedure will encompass the training of the first and second stage of classification, with the algorithm progression being multi-layer perceptron classifier, multi-layer perceptron regressor, random forest and support vector machine. For the first stage of classification, given the identical characteristics of each binary classifier, the hyperparameter tuning will be based on the first class classifier, with the results for every other binary classifier being presented at the end of each section. The second stage of classification will evaluate the impact of each type of input, along with its own hyperparameters.

As described in chapter 3, F1-score will be used in the assessment of each technique's impact to the overall performance of the algorithms. Given the vast possible combination of hyperparameters and architectures, the search for the best combination will be conducted in sequential manner, where the best performing parameter is chosen in each step of the procedure. While this method does not account for the possible interactions between different hyperparameters and it is likely to lead to a local maximum, it is the only feasible method within the time and computational constraint.

While the effects of the training procedure will only be visible through tables and graphs in the course of the next subsections, real world applications of the chosen architecture will be displayed at the end of the chapter along with an overall analysis.

In consideration of the section guideline follows the first algorithm test.

4.2.1 Multi-layer perceptron classifier

The first tested algorithm, multi-layer perceptron classifier, is a type of feed forward neural network capable of producing well defined outputs. For the purpose of the first stage of classification, the algorithm will be used in the binary classifiers as a first stage approach to hard classification.

Given the algorithm's possible architecture structures, the construction of the base classifier will start with only one season of inputs with the exclusion of the auxiliary indices, in turn, establishing a base input/output configuration of 12 to 1, respectively. The architecture, which will function as a base for all the subsequent alterations, will then feature a 12 node input layer, two hidden layers with 20 and 10 hidden nodes, respectively, and one node output layer, as seen in figure 4.1.

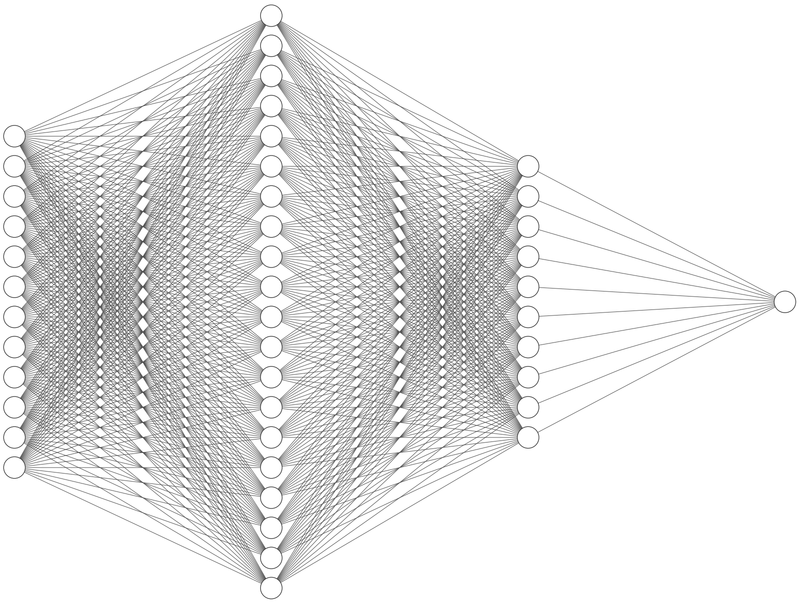


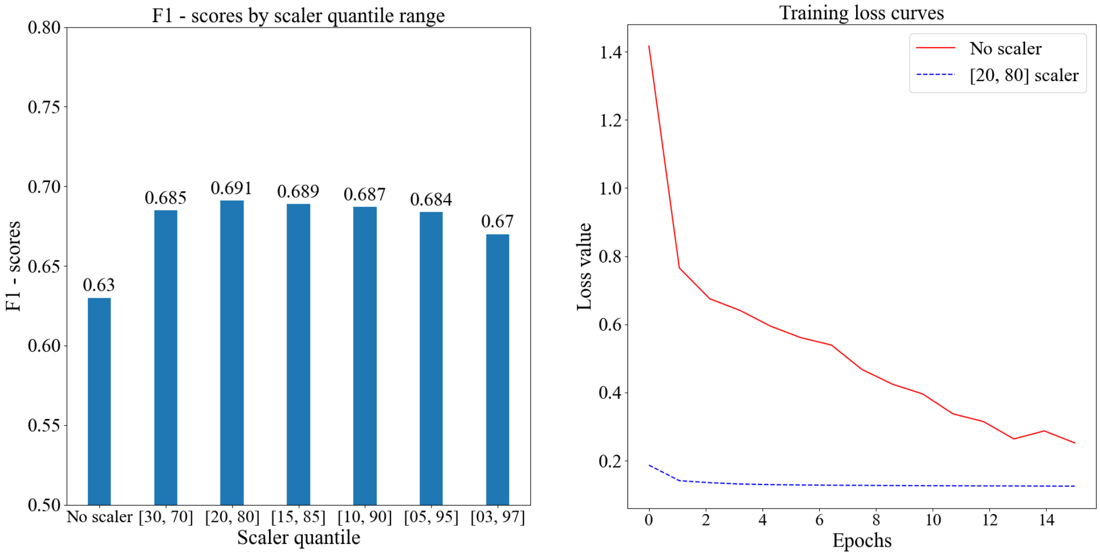
Figure 4.1: Starting multi-layer perceptron classifier architecture.

Through the aforementioned base architecture it will be possible to follow the first class classifier (i.e. *Artificialized*) progress and evaluate the impact of the prospective techniques. For this purpose, follows the initial results gathered from feeding the unprocessed data from a training subset containing 750.000 data points to the classifier, with the default hyperparameters: "relu"activation function, "adam"solver, "0.001"initial learning rate and "constant"learning rate. The presented values in each test were derived from the average values of 5 runs.

Table 4.2: Initial artificialized binary classifier scores in MLP classifier.

Classifier	Scores		
	Recall	Precision	F1
Artificialized	0,512	0,818	0,630

With the base values for the classifier defined in table 4.1, the testing and optimization procedure can begin. The first technique tested was the regularization of the input values. This technique centers and scales the input values around the value of 0, aiming to prevent values from different input sources to disproportionately affect the algorithm's computations, as such, a scaler entity (i.e. a scaling matrix) is necessary in order to to correctly access each input interval of operation. As previously mentioned in chapter 3, the regularization technique mainly focus on helping the convergence of the algorithm, however, with additional tuning it can help reduce the impact of eccentric values present in the data set. Given the high variance of values present in each spectral band, derived from the vastly different radiation absorption properties of each class, the delineation of eccentric values from regular ones represents an harder problem. Consequently, a tunable scaler capable of defining the desired quantile range was necessary, thus, the module "RobustScaler"acquired from the scikit-learn package was used.



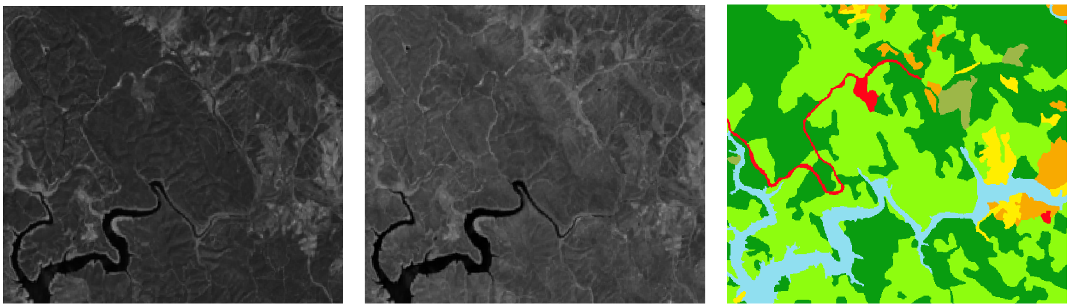
(a) F1 - scores by scaler quantile range.

(b) Loss convergence comparison.

Figure 4.2: Regularization test in MLP classifier.

The test featured the training of the algorithm with increasing scaler ranges, as seen in figure 4.2a. From the acquired results it is possible to confirm the effect of the regularization technique on the algorithm peak F1 - score, with increase of up to 10% peak score in comparison with the no scaler approach. In addition, as its primary function, the use of a scaler revealed drastic improvements in the convergence rate of the algorithm, observable in figure 4.2b. Consequently, a scaler with quantile range of 60% (i.e. [20, 80] interval) was chosen for subsequent implementations.

Following the implementation of the scaler, the selection of targeted spectral bands per classifier was tested. The technique aimed to reduce unnecessary complexity, by removing neutral or damaging spectral bands from each binary classifier. Given the vast amount of possible band combinations, the test started with the visual selection of the seemingly least useful band for a certain classifier.

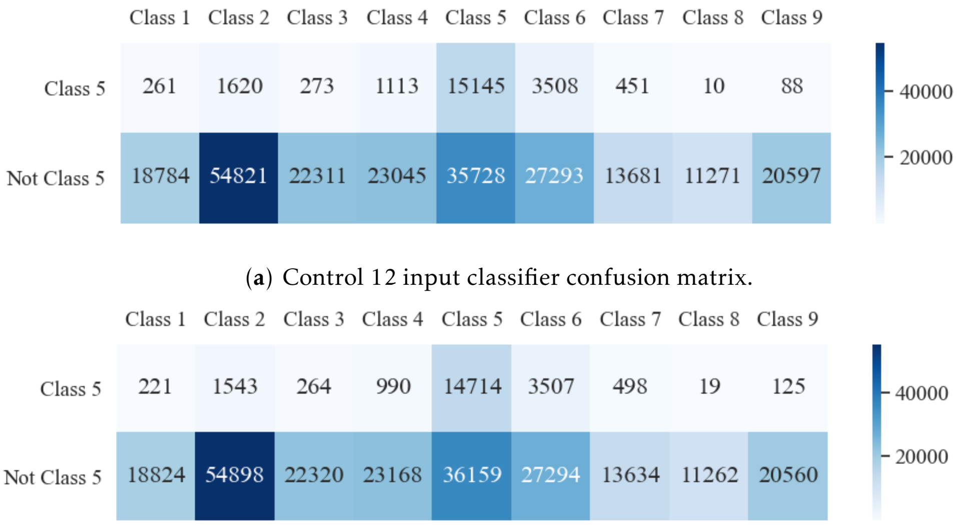


(a) Spectral band number 5. (b) Spectral band number 6. (c) Zone simplified COS label.

Figure 4.3: Band utility visual evaluation.

The search concluded with the selection of the spectral band number 5, represented in figure 4.3a, to be excluded from the original training data set of the *Forest* binary classifier, for the seemingly unrecognizable difference between the *Forest* major class and *Bushes* major class, represented by dark green and light green in figure 4.3c, respectively. Difference recognizable, as an example, in the spectral band number 6, figure 4.3b.

With the band selected and the classifier architecture adjusted for 11 inputs follows the acquired confusion matrices for the altered and control group test.



	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 5	261	1620	273	1113	15145	3508	451	10	88
Not Class 5	18784	54821	22311	23045	35728	27293	13681	11271	20597

(a) Control 12 input classifier confusion matrix.

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 5	221	1543	264	990	14714	3507	498	19	125
Not Class 5	18824	54898	22320	23168	36159	27294	13634	11262	20560

(b) Altered 11 input classifier confusion matrix.

Figure 4.4: Confusion matrices comparison in MLP classifier.

The analysis of the confusion matrices, depicted in figure 4.4, reveal, in accordance with the visual assessment, no significant difference in the miss classification of the *Bushes* major class as *Forest* major class, class 5 and 6, respectively. However, a slight decrease in the correct classification of the *Forest* major class (i.e. class 5) can be observed. Consequently, given the unpredictable overall negative effects of band removal, it was agreed to preserve every available band for subsequent implementations.

The next step in the training procedure involves the addition of auxiliary vegetation indices to the training data set. As previously described in chapter 3, the method aims to increase the algorithm's performance by mitigating the inability to derive information from all possible data transformations (i.g. division of input data). Consequently, follows the result of the implementation of each vegetation index to each classifier along with the result of the addition of all vegetation indices in an adjusted architecture with 19 nodes in the input layer, 30 hidden nodes in the first hidden layer, 15 hidden nodes in the second hidden layer and 1 node in the output layer.

Table 4.3: Vegetation Index impact analysis on MLP classifier.

Classifier	Control F1 - Scores	F1 - score improvements by index addition				
		Index 1 (NDVI)	Index 2 (MNDWI)	Index 3 (MNDWI2)	Index 4 (NDMI)	Index 5 (NDBI)
1. Artificialized	0,68	0	0	0	+ 0,01	0
2. Agricultural	0,57	- 0,01	0	- 0,03	0	0
3. Pasture	0,36	- 0,02	0	0	- 0,02	- 0,02
4. Agroforestry	0,43	0	+ 0,01	- 0,03	- 0,02	- 0,02
5. Forest	0,41	+ 0,04	+ 0,02	+ 0,03	+ 0,02	+ 0,03
6. Bushes	0,24	+ 0,04	0	+ 0,03	+ 0,03	- 0,03
7. Uncultured	0,41	- 0,02	- 0,01	+ 0,01	- 0,04	0
8. Wetlands	0,68	0	0	0	0	0
9. Water bodies	0,72	0	0	0	+ 0,01	+ 0,01

Table 4.4: Vegetation Index impact analysis on MLP classifier. (continuation)

F1 - score improvements by index addition		
Index 6 (BSI)	Index 7 (NDGI)	Combined Indices
0	+ 0,01	+ 0,02
- 0,02	- 0,01	+ 0,02
+ 0,02	+ 0,02	- 0,02
+ 0,01	- 0,01	+ 0,02
0	0	+ 0,03
+ 0,04	- 0,03	+ 0,06
- 0,03	- 0,01	+ 0,01
0	0	0
+ 0,01	0	+ 0,01

The obtained results, visible in table 4.3 and 4.4, suggest little impact to each classifier performance across every individual index addition, with the obtained values within the range of the characteristic variation derived from the random initiation of ANN weights. In contrast, the addition of every index to input layer appears to lead to a small increase in performance across most major class classifiers. Considering the negligible impact of the increase of input features to the actual classification execution time, the addition of vegetation indices, despite not demonstrating substantial impact in the achieved performance, shall be used in subsequent implementations.

Following the integration of the vegetation indices to the input layer, the next step in the training procedure is the analysis of a multi-temporal classification approach. To this end, three sets of tests were conducted in increasing number of encompassed seasons, each with equivalent algorithm architecture sizes, as seen in figure 4.5.

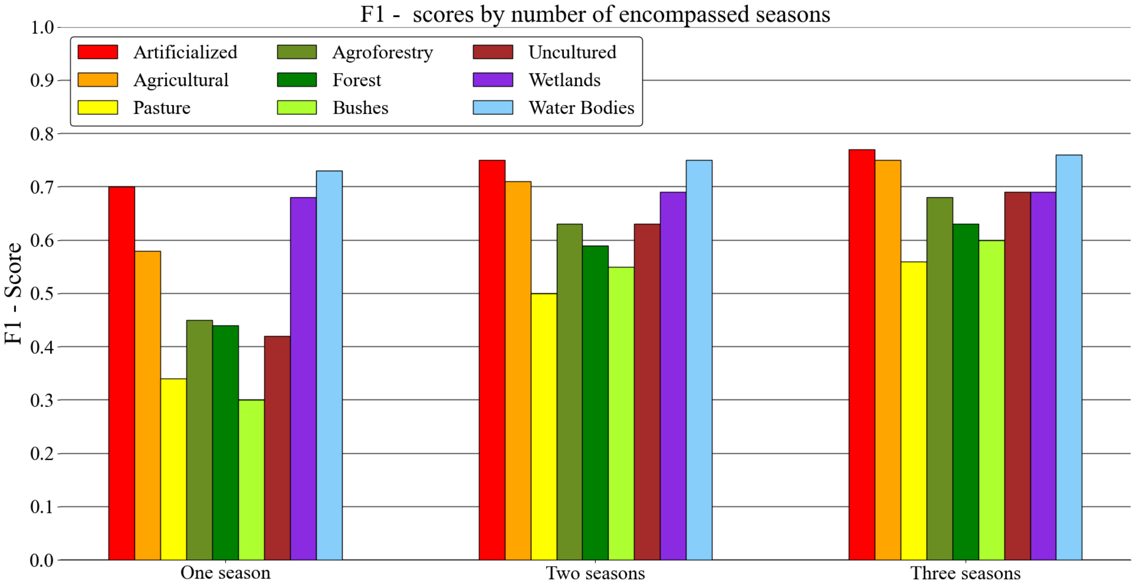


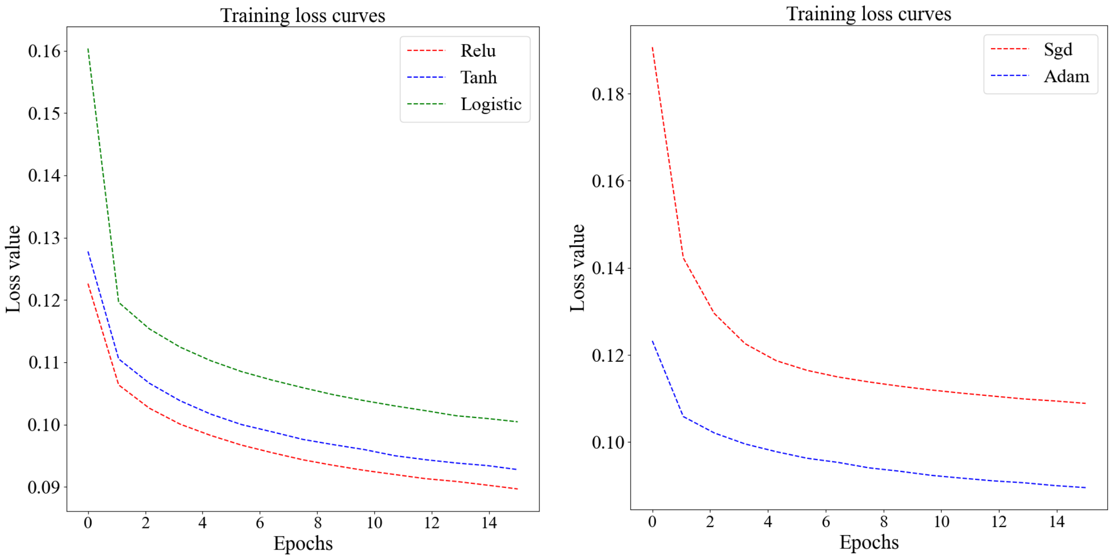
Figure 4.5: F1 - score assessment of multi-temporal approach in a MLP classifier.

Table 4.5: Multi-temporal approach analysis in a MLP classifier.

Classifier	F1 - score improvements relative to single temporal approach (%)		F1 - score variation from index addition
	Two seasons	Three seasons	
1. Artificialized	+ 7,1	+ 10,0	0
2. Agricultural	+ 22,4	+ 29.3	+ 0,01
3. Pasture	+ 47,1	+ 64,7	+ 0,01
4. Agroforestry	+ 40,0	+ 51,1	+ 0,04
5. Forest	+ 34,1	+ 43,2	0
6. Bushes	+ 83,3	+ 100,0	+ 0,02
7. Uncultured	+ 50,0	+ 64,3	+ 0,02
8. Wetlands	+ 1,5	+ 1,5	0
9. Water bodies	+ 2,7	+ 4,1	0

The analysis of the achieved multi-temporal results, as described in table 4.5, reveal vast improvements in the classification performance across every major classifier, with special impact in flora based classifiers (i.e. *Agricultural*, *Pasture*, *Agroforestry*, *Forest*, and *Bushes*), where improvements of up to 100% are observed. In turn, validating the multi-temporal approach as a suitable technique for land cover classification. Conversely, the addition of auxiliary vegetation indices in a multi-temporal approach, in accordance with the results gathered from the initial approach, appears to have limited impact in the distinction of most major classes. Nevertheless, the effect of the additional selected vegetation indices, when combined, have never been registered as detrimental to the overall performance of the classifier, thus, justifying its use in the final implementation.

The training procedure of the Multi-layer perceptron classifier concludes with the tuning of the inner architecture and respective hyperparameters. Consequently, the output of the *Artificialized* binary classifier is evaluated using various hidden layer sizes, activation functions, and solvers.



(**a**) Loss convergence per activation function.

(**b**) Loss convergence per solver.

Figure 4.6: Hyper parameter test in MLP classifier.

Table 4.6: Classifier architecture testing in MLP classifier.

Classifier Architecture	F1 - score
(40, 20)	0,770
(50, 30)	0,769
(70, 30)	0,773
(90, 45)	0,771
(100, 55)	0,778
(70, 70)	0,777
(80, 80)	0,779
(90, 90)	0,776

The tuning procedure started with the investigation of different inner architecture sizes from which the values present in table 4.6 were gathered. The values in the classifier architecture column represent the number of nodes present in the first and second hidden layers, with the test conducted exploring symmetrical and asymmetrical designs in ascending order. The analysis of the achieved results reveal little F1 - score variance between each architecture, with the small variance attributable to the increased number of nodes. The increase in F1 - score with the size of the artificial neural network, however, is accompanied by an increase in complexity, training time and likelihood of overfitting. As such, the architecture (70, 30) was selected for the final implementation for its balance between size and F1 - score. The second step in the tuning procedure was related with the selection of the activation function and solver. From the available options within the scikit-learn package, the most commonly used hyperparameters were selected and the resulting loss curves compared. As visible in figure 4.6, the rectified linear unit function "ReLU", and "Adam", a stochastic gradient-based optimizer proposed by Kingma, Diederik, and Jimmy Ba [158], produced the fastest convergent loss curves in their respective areas, consequently, ReLU activation function and Adam solver were selected for the final algorithm implementation.

Finalized the training procedure for the first stage of classification of the Multi-layer perceptron classifier, the algorithm was able to achieve a balanced overall F1 - score of 68% across the employed data set. The acquired accuracy values, as seen in table 4.7, shall now be used as the benchmark for the efforts of the second stage of classification.

Table 4.7: Final multi-layer perceptron classifier results

Classifier	F1 - score	Recall	Precision
1. Artificialized	0,772	0,698	0,862
2. Agricultural	0,750	0,715	0,787
3. Pasture	0,560	0,462	0,711
4. Agroforestry	0,676	0,650	0,704
5. Forest	0,634	0,558	0,735
6. Bushes	0,598	0,518	0,709
7. Uncultured	0,686	0,622	0,766
8. Wetlands	0,694	0,982	0,536
9. Water bodies	0,765	0,634	0,965

4.2.2 Multi-layer perceptron regressor

The second tested algorithm, multi-layer perceptron regressor, is type of feed forward network capable of producing continuous outputs. For the purpose of the first stage of classification, the algorithm will be tested in the binary classifiers as a first stage approach to soft classification with the aim of providing additional information about the likelihood, or confidence level, of each binary classifier decision as a non-integral input for the second stage of classification. Within the sci-kit learn package, the algorithm is of identical structure to the previously tested algorithm, differing in comparison to the MLP classifier by the absence of an activation function in the output layer and, as a result, a distinct loss function (i.e. from a logistic loss function in the MLP classifier algorithm to a square error loss function in the MLP regressor). Consequently, given the interchangeable hyperparameters employed by both algorithms, the training procedure of the MLP regressor can be excluded as the obtained values will be equivalent.

Given the continuous output format of the aforementioned algorithm, the previously applied performance assessment methods (i.g. F1 - score) can not be computed without the presence of a threshold to define the output labels. Although the defined threshold may not describe the optimal interpretation of the predicted values and will be not be present in the final implementation, it is necessary to provide comparing values. As such, for the purpose of the training assessment procedure, follows the final F1 - scores acquired through the application of a standard decision threshold of 0 to the output layer,as described in table 4.8.

Table 4.8: Final multi-layer perceptron regressor results.

Classifier	F1 - score	Recall	Precision
1. Artificialized	0,775	0,706	0,857
2. Agricultural	0,734	0,672	0,808
3. Pasture	0,560	0,458	0,723
4. Agroforestry	0,658	0,601	0,726
5. Forest	0,636	0,546	0,761
6. Bushes	0,581	0,480	0,735
7. Uncultured	0,650	0,552	0,790
8. Wetlands	0,694	0,981	0,536
9. Water bodies	0,763	0,630	0,967

4.2.3 Random forest classifier

The third tested algorithm, Random forest classifier, is an increasingly popular ensemble learning method based on the construction of a large number of different decision trees. For the purpose of the first stage of classification, the algorithm, similarly to the multi-layer perceptron classifier, will be employed as a first stage approach to hard classification.

In accordance with the with the previously followed training procedure, the random forest algorithm will initially use the base 12 inputs of a single season with default hyperparameters. From the defined base architecture, each technique will be applied and subsequent results analysed. Given the higher sensibility displayed by other major classes in previous tests in comparison with the previously targeted major class (i.e. *Artificialized* class), the assessment of each technique impact in subsequent tests will follow the progress of the *Bushes* major class (i.e. class 6). Thus, follows the initial results gathered from feeding the unprocessed data from the previously employed training subset containing 750.000 data points to the classifier, with the default hyperparameters: "gini"criterion, "100"n_estimators (i.e. number of trees), "2"min_samples_split with no maximum depth. The presented values in each test were derived from the average values of 5 runs.

Table 4.9: Initial Bushes binary classifier scores in RF classifier.

Classifier	Scores		
	Recall	Precision	F1
Bushes	0.185	0.713	0.294

Similarly to the values obtained in the first run of the MLP classifier, observable in the left most bar cluster in figure 4.5, the *Bushes* major class classifier is incredibly weak when provided with only the raw data and no further adjustments, correctly classifying only 18,5% of the existing positive instances. Consequently, considering the base values displayed in table 4.9, follows the achieved results from the implementation of the first 2 techniques, scaler and index addition.

Table 4.10: Scaler and vegetation indices impact analysis on Random forest classifier.

Classifier	Control F1 - Scores	Relative F1 - score change by technique addition	
		Scaler ([80, 20])	Combined indices
1. Artificialized	0,66	0	0
2. Agricultural	0,58	- 0,02	- 0,02
3. Pasture	0,38	0	- 0,01
4. Agroforestry	0,36	0	+ 0,01
5. Forest	0,44	0	- 0,01
6. Bushes	0,29	+ 0,01	0
7. Uncultured	0,35	+ 0,01	+ 0,03
8. Wetlands	0,69	0	0
9. Water bodies	0,73	0	0

In contrast with the increase in performance of up to 10% observed in the implementation of the scaler technique to the MLP classifier first class, verified in figure 4.2a, the implementation of a scaler to the input data of the random forest algorithm has demonstrated no tangible improvement in any of the reviewed major classes. Likewise, in contrast with the MLP test, the addition of the vegetation indices to the classifier has led to an overall decrease in the classifier performance, as seen in table 4.10, in turn, challenging the usefulness of vegetation indices in random forest classifiers. As a result, while the implementation of vegetation indices for multi-temporal impact assessment continues, the use of a scaler will be absent in subsequent implementations.

Following the incorporation of the vegetation indices and scaler to the input layer, the next step in the training procedure is the analysis of the multi-temporal classification approach. In accordance with the previously applied testing procedure, three sets of test were conducted with increasing number of encompassed seasons and vegetation indices.

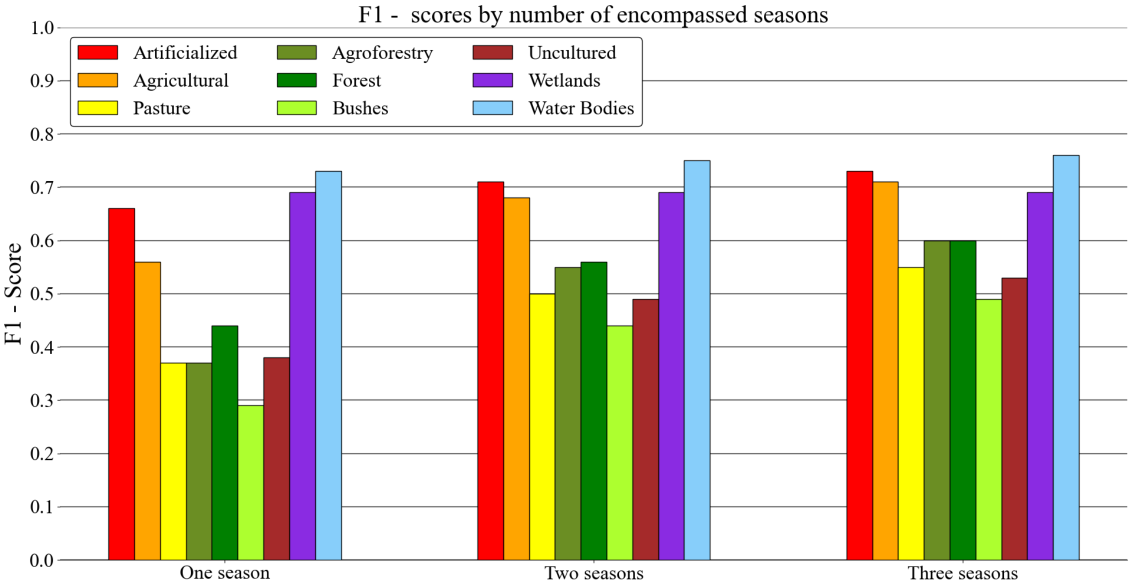


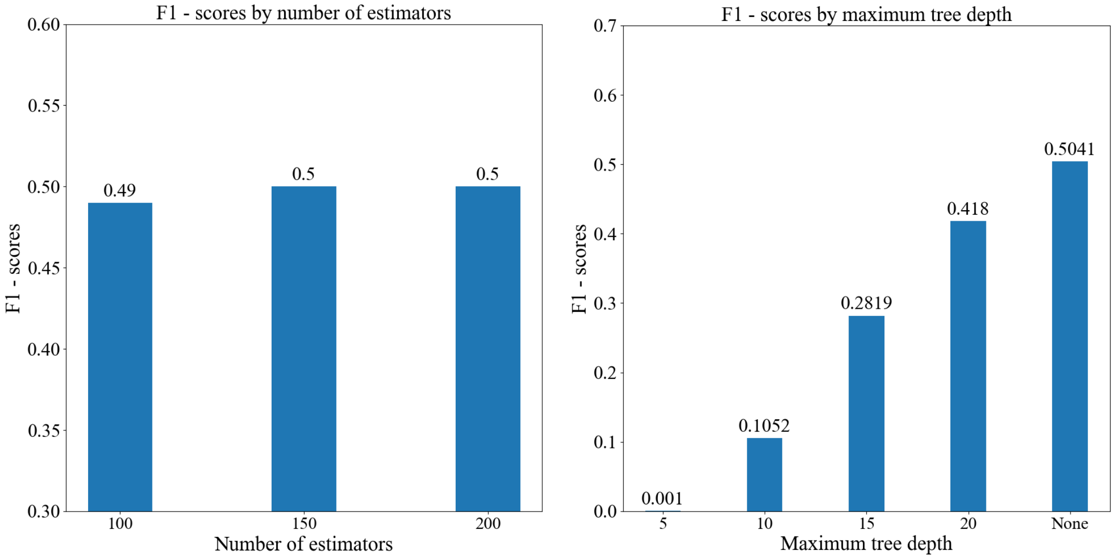
Figure 4.7: F1 - score assessment of multi-temporal approach in a RF classifier.

Table 4.11: Multi-temporal approach analysis in a RF classifier.

Classifier	F1 - score improvements relative to single temporal approach (%)		F1 - score variation from index addition
	Two seasons	Three seasons	
1. Artificialized	+ 7,6	+ 10,1	0
2. Agricultural	+ 21,4	+ 26.8	- 0,01
3. Pasture	+ 35,1	+ 48,6	0
4. Agroforestry	+ 48,6	+ 62,2	- 0,02
5. Forest	+ 27,3	+ 36,4	- 0,01
6. Bushes	+ 51,7	+ 69,0	- 0,01
7. Uncultured	+ 29,0	+ 39,5	0
8. Wetlands	0	0	0
9. Water bodies	+ 2,7	+ 4,1	0

With exception of the *Wetlands* and *Water bodies* major classes which remain relatively unaffected, the study of the multi-temporal method shows, in accordance with the previously tested algorithms, significant improvements in the performance of each binary classifier across the number of encompassed seasons, as seen in figure 4.7 and table 4.11. The acquired results however, despite validating the use of multi-temporal approaches with random forest algorithms, exhibit lower absolute values when compared to the MLP classifier approach. Furthermore, in contrast with the aforementioned algorithm, the inclusion of auxiliary vegetation indices appears to be detrimental to the performance of the RF classifier. As a result, future implementations of the RF algorithm will not employ auxiliary vegetation indices.

The training procedure of the random forest classifier concludes with the tuning of some of its respective hyperparameters. Consequently, the output of the *Bushes* binary classifier is evaluated using varying number of estimators, criterion and max depth levels.



(**a**) F1 - score by number of estimators.

(**b**) F1 - score by maximum tree depth.

Figure 4.8: Hyper parameter test in RF classifier.

Table 4.12: Criterion testing.

Criterion	F1 - score
Gini	0,501
Entropy	0,507

Table 4.13: Maximum number of features per split testing.

Maximum number of features per split	F1 - score
Square root	0,503
Binary logarithm	0,494
All features	0,555

The tuning procedure started with the exploration of the optimal number of estimators (i.e. number of decision trees from which the classifier is assembled), the test started with the default value of 100, and proceeded in increments of 50 estimators. In accordance with previous applications described in the literature [119], the test of the optimal number of estimators, visible in figure 4.8a, has shown no improvements in the performance of the classifier with the increase in the number of estimators. The following test explored the optimal maximum tree depth, the test aimed to identify a limited depth level in order to reduce the potential of overfitting, however, as displayed in figure 4.8b, every attempt at limiting the maximum level was detrimental in the classification of the new, unseen data of the testing data set. As such, in light of the aforementioned tests, the standard number of estimators "100"was selected along with an uncapped maximum tree depth for the final RF implementation. The tuning procedure concluded with the analysis of the best function to measure the quality of a split (i.e. Criterion) and the maximum number of features to consider in each split. As visible in table 4.12, the criterion test revealed a slight increase in performance when the criterion "Entropy"was employed in place of the criterion "Gini". Similarly, the availability of all features as split parameters in each tree node has resulted in an increase of up to 10% in the *Bushes* classifier performance when compared with the square root or binary logarithm of the total number of features, findings displayed in table 4.13. Consequently, the "Entropy"criterion and the total number of features per split will be selected as the optimal hyperparameters for their respective categories for the final RF implementation.

Finalized the training procedure for the first stage of classification of the Random forest classifier, with the values described in table 4.14, the algorithm was able to achieve a balanced overall F1 - score of 66,4% across the employed data set, result slightly lower than the 68% overall F1 - score achieved by the Multi-layer perceptron classifier.

Table 4.14: Final random forest classifier results.

Classifier	F1 - score	Recall	Precision
1. Artificialized	0,745	0,647	0,878
2. Agricultural	0,735	0,666	0,821
3. Pasture	0,575	0,452	0,790
4. Agroforestry	0,662	0,585	0,761
5. Forest	0,633	0,528	0,792
6. Bushes	0,562	0,430	0,807
7. Uncultured	0,594	0,463	0,827
8. Wetlands	0,695	0,981	0,538
9. Water bodies	0,760	0,622	0,978

4.2.4 Support vector machine classifier

The fourth and last tested algorithm, Support vector machine classifier, is a supervised learning model based on the division of the feature space through the use of an hyperplane. For the purpose of the first stage of classification, the algorithm, similarly to the multi-layer perceptron and random forest classifiers previously tested, will be employed as a first stage approach to hard classification.

Following the previously applied training procedure, the algorithm will be initialized with 12 inputs of a single season, the default hyperparameters however, displayed the inability to successfully create a base from which the classification procedure for most major classes would be possible. For the present case, the default package value for the regularization parameter C (i.e. default value of 1), failed to successfully categorize most classes, consequently an arbitrary higher value of 25 was initially selected. Furthermore, given the quadratic increase in computation time with the addition of data samples, the SVM algorithm, in similarity with the random forest classifier, was trained through the use of the ensemble bagging technique. Method in which multiple smaller SVM algorithms are trained using smaller random data subsets, with subsequent results defined via a voting system. In addition, in order to prevent certain features from dominating the feature space and ease the kernel calculations, the implementation of a scaler is essential from the beginning.

In accordance with the previously tested algorithms, each technique will be applied to the defined base architecture and respective results analysed through the impact on the *Bushes* major class binary classifier (i.e. class 6).

Thus, follows the initial results gathered from feeding the SVM classifier with data from a training subset containing 500.000 data points with a bagging ration of 1:8 and hyperparameters: "radial basis function"kernel, "25"regularization parameter C, and "scale"kernel coefficient γ. The presented values in each test were derived from the average values of 5 runs.

Table 4.15: Initial Bushes binary classifier scores in SVM classifier.

Classifier	Scores		
	Recall	Precision	F1
Bushes	0.031	0.723	0.059

When given only the scaled data from one season, the *Bushes* SVM classifier, despite similarities in the trend demonstrated by the previously tested algorithms, has shown remarkably bad results, correctly classifying only 3,1% of the existing positive instances. Consequently, considering the base values displayed in table 4.15, follows the achieved results from the implementation of the first technique, index addition.

Table 4.16: Vegetation indices impact analysis on SVM classifier.

Classifier	Control F1 - Scores	Relative F1 - score change by technique addition
		Combined indices
1. Artificialized	0,61	+ 0,02
2. Agricultural	0,47	0
3. Pasture	0,12	- 0,01
4. Agroforestry	0	0
5. Forest	0,27	+ 0,01
6. Bushes	0,06	0
7. Uncultured	0,19	+ 0,01
8. Wetlands	0,66	+ 0,01
9. Water bodies	0,70	+ 0,01

The analysis of the values gathered in table 4.16, describe the SVM classifier as the lowest performing classifier among all tested algorithms in single temporal approaches. The algorithm was unable to derive an hyper-plane capable of classifying the *Agroforestry* major class and failed to achieve the 0,30 F1 - score threshold in the majority of the binary classifiers. Furthermore, despite positive, the impact of the auxiliary vegetation indices was not sufficient to increase the performance of each classifier to comparable levels with the previously tested algorithms. In turn, challenging the application of SVM algorithms in single temporal land use and land cover implementations based on spectral reflectance.

In accordance with the previously applied testing procedure, follows the analysis of the multi-temporal classification approach through the creation of three sets of test composed by increasing numbers of encompassed seasons and vegetation indices.

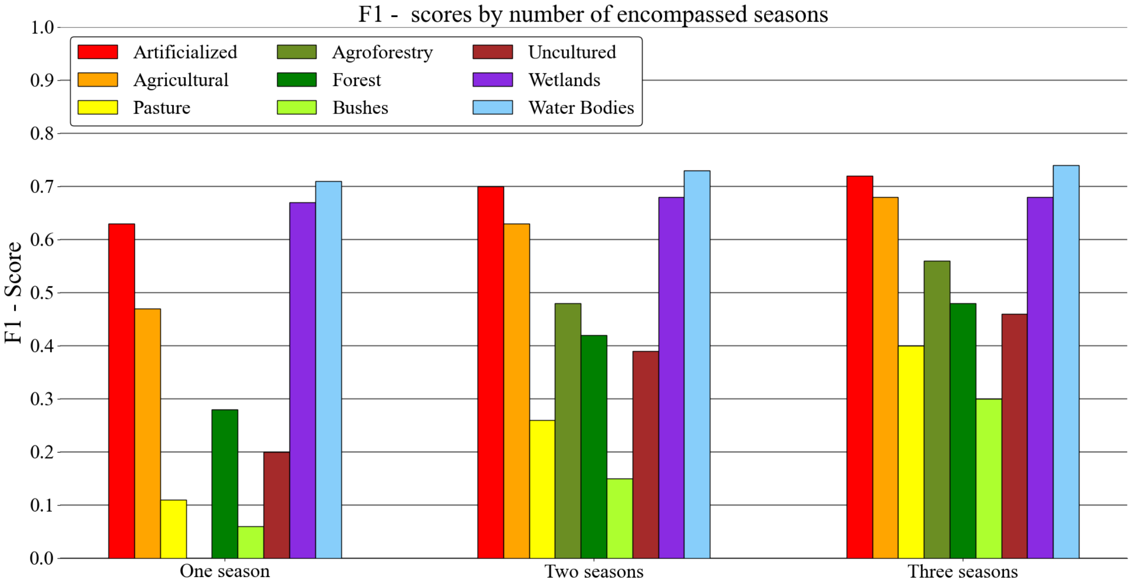


Figure 4.9: F1 - score assessment of multi-temporal approach in a SVM classifier.

Table 4.17: Multi-temporal approach analysis in a SVM classifier.

Classifier	F1 - score improvements relative to single temporal approach (%)		F1 - score variation from index addition
	Two seasons	Three seasons	
1. Artificialized	+ 11,1	+ 14,2	0
2. Agricultural	+ 34,0	+ 44,7	+ 0,01
3. Pasture	+ 136,4	+ 263,6	+ 0,02
4. Agroforestry	NaN	NaN	+ 0,02
5. Forest	+ 50,0	+ 71,4	+ 0,02
6. Bushes	+ 150,0	+ 400,0	+ 0,09
7. Uncultured	+ 95,0	+ 130,0	+ 0,02
8. Wetlands	+ 1,4	+ 1,4	0
9. Water bodies	+ 2,8	+ 4,2	0

In line with the previously tested algorithms, the SVM classifier shows enormous improvements in every flora based classifier with the increase in the number of encompassed seasons. Ultimately allowing the classification of the previously indistinguishable *Agroforestry* major class, as seen in figure 4.9. Furthermore, the addition of auxiliary vegetation indices to the multi-temporal approach resulted in the largest absolute performance increase across the reviewed classifiers, as described in table 4.17, in turn, validating the use of vegetation indices in subsequent implementations. The training procedure of the support vector machine classifier concludes with the tuning of the two primary hyperparameters in implementations based on the radial basis function. Accordingly, follows the evaluation of the C regularization parameter in the *Pasture* and *Agroforestry* binary classifiers and the impact of the gamma hyperparameter in the *Bushes* major class.

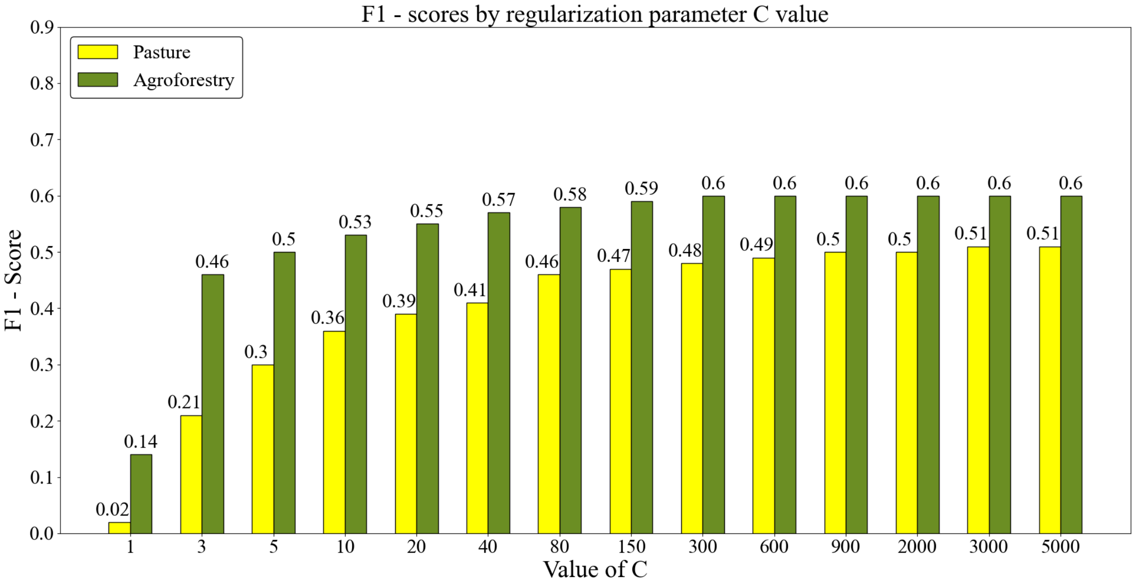


Figure 4.10: F1 - score progression through regularization parameter C value.

Table 4.18: Classifier gamma testing in SVM classifier.

Gamma	F1 - score
$\frac{1}{number\,of\,features \cdot feature\,variance}$	0,510
$\frac{1}{number\,of\,features}$	0,482

The tuning procedure began with an assessment of the optimal regularization parameter C for different SVM binary classifiers. The study looked at the progression of F1 - scores for the *Pasture* and *Agroforestry* binary classifiers as C values increased, revealing, as observable in figure 4.10, high variance in the performance of each classifier until a threshold is hit, highlighting the importance of an appropriate regularization parameter. Furthermore, the test demonstrated a direct relationship between the magnitude of the required C value and the relative difficulty (i.e. overlap between multiple classes, defined by small nuances) of the targeted class. As an example, the *Agroforestry* classifier is observed to plateau around the C value of 300, whereas the *Pasture* classifier continues to increase until the C value of 3000 is reached. As a result, the optimization of the regularization parameter C differs for each classifier, revealing the standard package value of 1 and then selected arbitrary value of 25 as insufficient for an appropriate classification.

Consequently, for the final SVM classifier implementation, each regularization parameter C was tuned separately.

The tuning procedure concluded with the analysis of the best kernel coefficient γ value. The test employed the available standard γ modes "scale"and "auto", computed by the equations in the top and bottom rows of table 4.18, respectively, in the training procedure of the *Bushes* classifier. The analysis of the achieved results reveal a decrease in the performance of the classifier of 5,5% when the "auto"γ was employed, in turn, consolidating the standard value as the selected value for the final SVM implementation.

Finalized the training procedure for the first stage of classification of the Support vector machine classifier, with the results described in table 4.19, the algorithm was able to achieve a balanced overall F1 - score of 62,7% across the employed data set with optimized regularization parameter C. The acquired result were slightly lower than the 66,4% and 68% overall F1 - scores achieved by the Random forest classifier and Multi-layer perceptron classifier, respectively.

Table 4.19: Final support vector machine classifier results.

Classifier	F1 - score	Recall	Precision	Optimized C
1. Artificialized	0,727	0,639	0,845	300
2. Agricultural	0,704	0,627	0,804	600
3. Pasture	0,509	0,410	0,671	3000
4. Agroforestry	0,601	0,515	0,721	900
5. Forest	0,593	0,500	0,728	5000
6. Bushes	0,513	0,411	0,680	3000
7. Uncultured	0,582	0,490	0,717	3000
8. Wetlands	0,681	0,970	0,524	300
9. Water bodies	0,745	0,606	0,967	80

4.2.5 First stage conclusion

With the completion of the training procedure for each of the selected algorithms it was possible to extract information about the impact and potential benefits of a multi-temporal approach to the creation LCLU maps based on satellite multi-spectral imagery. The procedure explored the affinity of each algorithm to the addition of contextual information, along with the impact of the ever increasing vegetation indices. In summary, the incorporation of additional temporal data improved the performance, to varying degrees, of all of the reviewed algorithms. By contrast, the inclusion of vegetation indices in the input data revealed more erratic results, as evidenced by small increases in performance displayed by the MLP and SVM classifiers in contrast with small decreases in performance observed in the RF classifier. Lastly, while the RF classifier achieves the highest classification F-1 scores in the single-temporal implementation, the MLP classifier and MLP regressor achieve the overall highest classification scores in the multi-temporal implementation, comparison visible in tables 4.20 and 4.21.

Table 4.20: Final optimized first stage F1 - score comparison in single-temporal approach.

Classifier	F1 - score achieved in each major class								
	Artificialized	Agricultural	Pasture	Agroforestry	Forest	Bushes	Uncultured	Wetlands	Water bodies
MLP classifier	**0,696**	0,582	0,339	**0,451**	0,440	0,296	**0,424**	0,682	0,727
MLP regressor	0,695	0,558	0,372	0,428	0,436	0,310	0,420	0,683	0,727
RF classifier	0,672	**0,583**	**0,400**	0,416	**0,459**	**0,330**	0,391	**0,689**	**0,730**
SVM classifier	0,647	0,517	0,242	0,190	0,358	0,127	0,330	0,670	0,715

Table 4.21: Final optimized first stage F1 - score comparison in multi-temporal approach.

Classifier	F1 - score achieved in each major class								
	Artificialized	Agricultural	Pasture	Agroforestry	Forest	Bushes	Uncultured	Wetlands	Water bodies
MLP classifier	0,772	**0,750**	0,560	**0,676**	0,634	**0,598**	**0,686**	0,694	**0,765**
MLP regressor	**0,775**	0,734	0,560	0,658	**0,636**	0,581	0,650	0,694	0,763
RF classifier	0,745	0,735	**0,575**	0,662	0,633	0,562	0,594	**0,695**	0,760
SVM classifier	0,727	0,704	0,509	0,601	0,593	0,513	0,582	0,681	0,745

4.2.6 Second stage algorithm selection

With the conclusion of the first stage of classification, the second stage aims to aggregate each binary output from the first set of classifiers into one fully multi-class classifier. The separation of the aforementioned procedure into two stages allows for a modular increment of new classes or the dissolution of previous major classes in future implementations. Retraining of which, requires only the training of the added classes' binary classifiers along with the smaller and less computation demanding second stage classifier (i.e. 57 inputs in the first stage versus nº of classes as inputs in the second stage).

The selection and training of the algorithm for the second stage is relatively simple, thanks to the knowledge of the algorithms' hyperparameters gained in the first stage training procedure. As previously analysed, the MLP algorithm achieved the highest overall F1 - scores in both of its respective categories, hard and soft classification (i.e. classifier and regressor), with the RF algorithm achieving the highest results with limited data (i.e. in the single-temporal approach). As such, both MLP classifier and RF classifier will be tested for the second stage of classification, where each algorithm will be fed binary and continuous data from the previous per class classifiers, gathered through the use of the MLP classifier and MLP regressor, respectively. Both algorithms will employ the best combination of hyperparameters devised in previous tests, with the MLP classifier using the configuration of 12-15-7-1 (i.e, 12 nodes in the input layer, 15 nodes in the first hidden layer, 7 nodes in the second hidden layer and 1 output node) as the neural network architecture. For the final implementation, all first stage classifiers were retrained with a data subset containing 3.000.000 data points.

Table 4.22: Second stage MLP classifier testing.

Input data	Overall accuracy score
MLP classifier	0.686
MLP regressor	0.726

Table 4.23: Second stage RF classifier testing.

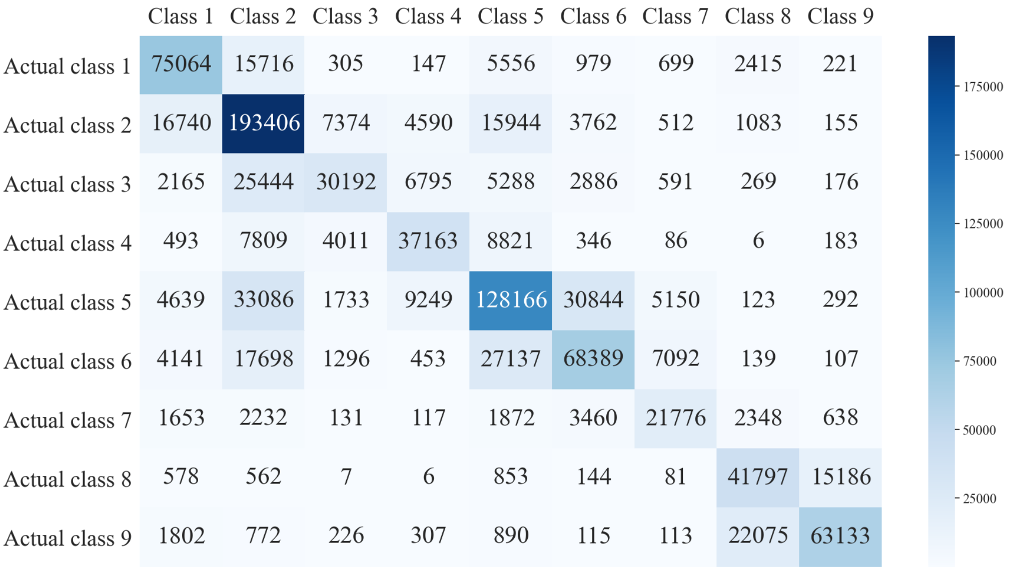
Input data	Overall accuracy score
MLP classifier	0.687
MLP regressor	0.727

An examination of the values in tables 4.22 and 4.23 reveal nearly identical results for each classifier in both types of input data. Conversely, while both algorithms struggle to improve the overall accuracy above the MLP classifier's previously calculated overall balanced F1 - score of 68 percent when using binary data, the ability to work with continuous data capable of supplying the second stage algorithm with information about the relative confidence level of each first stage classifier's decision has led to a noticeable increase in the final classification performance. Ultimately, despite the RF classifier's marginally better performance when compared to the MLP classifier, the extreme disparity in algorithm's storage requirements (i.e. 15KB for the MLP classifier and 5,7GB for the RF classifier), uphold the MLP classifier as the most suitable algorithm for the intended application.

In conclusion, follows the achieved overall accuracy and respective confusion matrix of the "out of sight"test executed with an unique data set containing 1.000.000 data points, described in table 4.24 and figure 4.11, respectively.

Table 4.24: Final out of sight overall accuracy.

Out of sight overall accuracy
0,659



	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Actual class 1	75064	15716	305	147	5556	979	699	2415	221
Actual class 2	16740	193406	7374	4590	15944	3762	512	1083	155
Actual class 3	2165	25444	30192	6795	5288	2886	591	269	176
Actual class 4	493	7809	4011	37163	8821	346	86	6	183
Actual class 5	4639	33086	1733	9249	128166	30844	5150	123	292
Actual class 6	4141	17698	1296	453	27137	68389	7092	139	107
Actual class 7	1653	2232	131	117	1872	3460	21776	2348	638
Actual class 8	578	562	7	6	853	144	81	41797	15186
Actual class 9	1802	772	226	307	890	115	113	22075	63133

Figure 4.11: Final out of sight confusion matrix.

4.2.7 Performance analysis

After assessing the final implementation accuracy, some conclusions can be drawn about the employed strategies.

- **The number of classifiable classes has a significant impact on LCLU classifiers:** as previously observed in the multiple per class F1 - score assessment tables, the difficulty of each class categorization task varies considerably with the targeted domain. Even in the final implementation results, visible in figure 4.11, the overlap between class 5 and class 6, *Forest* and *Bushes* respectively, is still prominent. As a result, it is common to see these classes aggregated into a single class in order to reduce complexity at the cost of functionality.

- **A multi temporal approach to LCLU mapping is advantageous:** the inclusion of additional temporally shifted spectral information to machine learning techniques, as previously documented throughout the training procedure, has resulted in the leverage of the classifiers' performance across all of the reviewed algorithms.

- **The effects of auxiliary vegetation indices were unremarkable:** numerous vegetation indices exist throughout the literature with varying applicability, and although certain indices help emphasise and improve the visual clarity of maps, the addition of the reviewed auxiliary vegetation indices to the tested algorithms in the particular implementation has failed to deliver on the improvements often described in the literature.

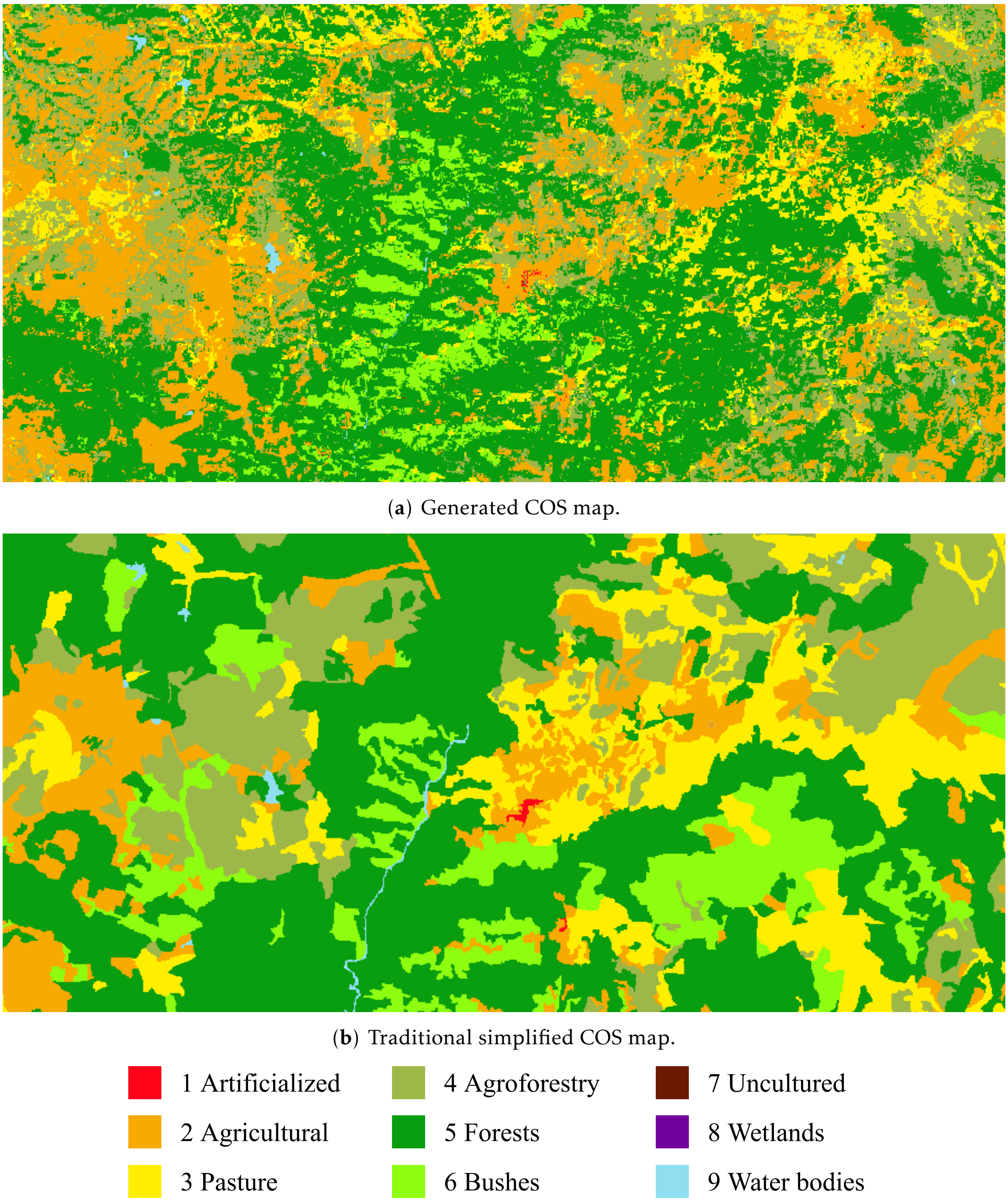
- **The volume of acquired data was insufficient:** given the computational limitation of individual testing and the absence of servers/supercomputers infrastructure, the acquired 6.425.998 land data points, although diverse, represent a mere 0,0072% of Portugal's 89.015 km^2 continental land area. Value drastically lower than the value of 0,25% advised in the reviewed literature as the optimal study area.

- **The implementation ground truth (i.e. the utilized COS map) and respective performance benchmark (i.e. accuracy assessment method) possess inherent errors:** given the scale of the intended application, many parasitic errors were accumulated throughout the required classification steps. Firstly, the expected human error factor in the creation of a nation wide LCLU map (i.e. original COS map) derived from distinct data sources, time zones, and human individual criteria. Secondly, temporal inconsistencies derived from the distinct time frame between the completion of the utilized COS map and the gathered raw satellite data, challenge inevitable in the intended year-round implementation. Lastly, and perhaps most importantly, the distinct classification approaches between the intended implementation and the used COS map. Which, given the minimum area requirement for classification of one hectare, is ultimately not a per-pixel approach, but rather object-based classification. All of the aforementioned factors have a significant impact on the developed classifier's perceived, portrayed, and actual performance.

4.3 In-plugin implementation

With the training procedure completed, the best performing algorithm can now be implemented in the QGIS plug-in. Since all of the data collection and pre-processing was codded on top of the QGIS framework from the beginning, little work was required to integrate the classification procedure. Given the square coordinates of the desired area, the plug-in automatically downloads, extracts and compiles all the relevant data from the sentinel-2 data base into a pandas dataframe where all subsequent vegetation indices are computed and attached. Next, all nine of the trained algorithms from the first classification stage are called to predict each class based on the assembled dataframe, the aforementioned procedure creates a new dataframe with all the per-class classifiers results. The new dataframe is then fed into the second stage algorithm to create the final classification vector with multi-class values (i.e. from 1 to 9). Finally, the multi-class vector is reconstructed as a matrix to create the final classification raster layer.

After completing the integration of the classification procedure, the developed per-pixel automatic classifier can now be visually compared with the traditional one on concrete images.



(a) Generated COS map.

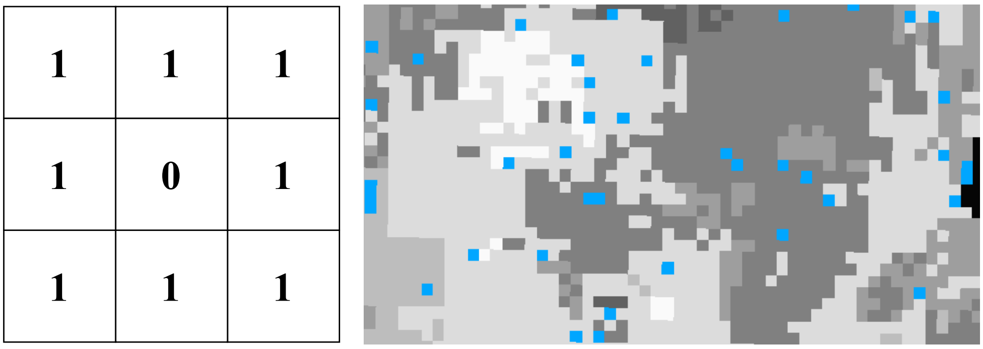
(b) Traditional simplified COS map.

(c) Class labels.

Figure 4.12: COS map comparison.

Despite the likely time difference between the creation of both images, derived from the multiple data sources of the traditional method [152] in comparison with the temporal consistent image set from which the generated COS map was assembled, the analysis of the provided images in figure 4.12 reveals many of the same features. In contrast, and as a result of differences in classification approach (i.e. Minimum mapping unit of one hectare in the conventional COS map as opposed to the per-pixel classification of the generated COS map), a lot more granularity can be observed in the generated COS map.

In order to reduce undesirable complexity and diminish the "salt and pepper"effect often present in per-pixel approaches, a final mask is applied to the generated image in order to remove singular encompassed pixels. The developed mask iterates once through every pixel and, with the exception of Artificialized pixels, replaces every encircled pixel with the class that has the majority of representation in the surrounding area, as exemplified in figure 4.13. The aforementioned technique contributes with an average of 1% increase in the classification accuracy across the reviewed images.



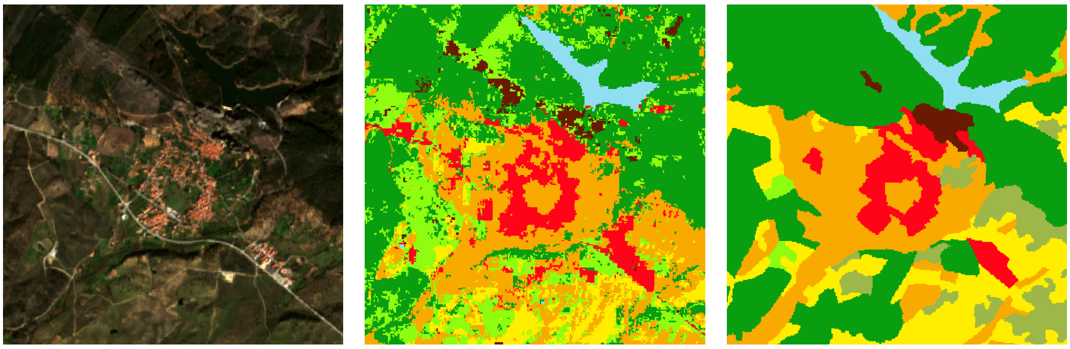
1	1	1
1	0	1
1	1	1

(a) Mask region.

(b) Example of encompassed pixels (light blue).

Figure 4.13: Region and application example of the constructed mask.

With all procedures in place, follows a direct comparison between the generated and the traditional simplified COS map.



(a) Land area.

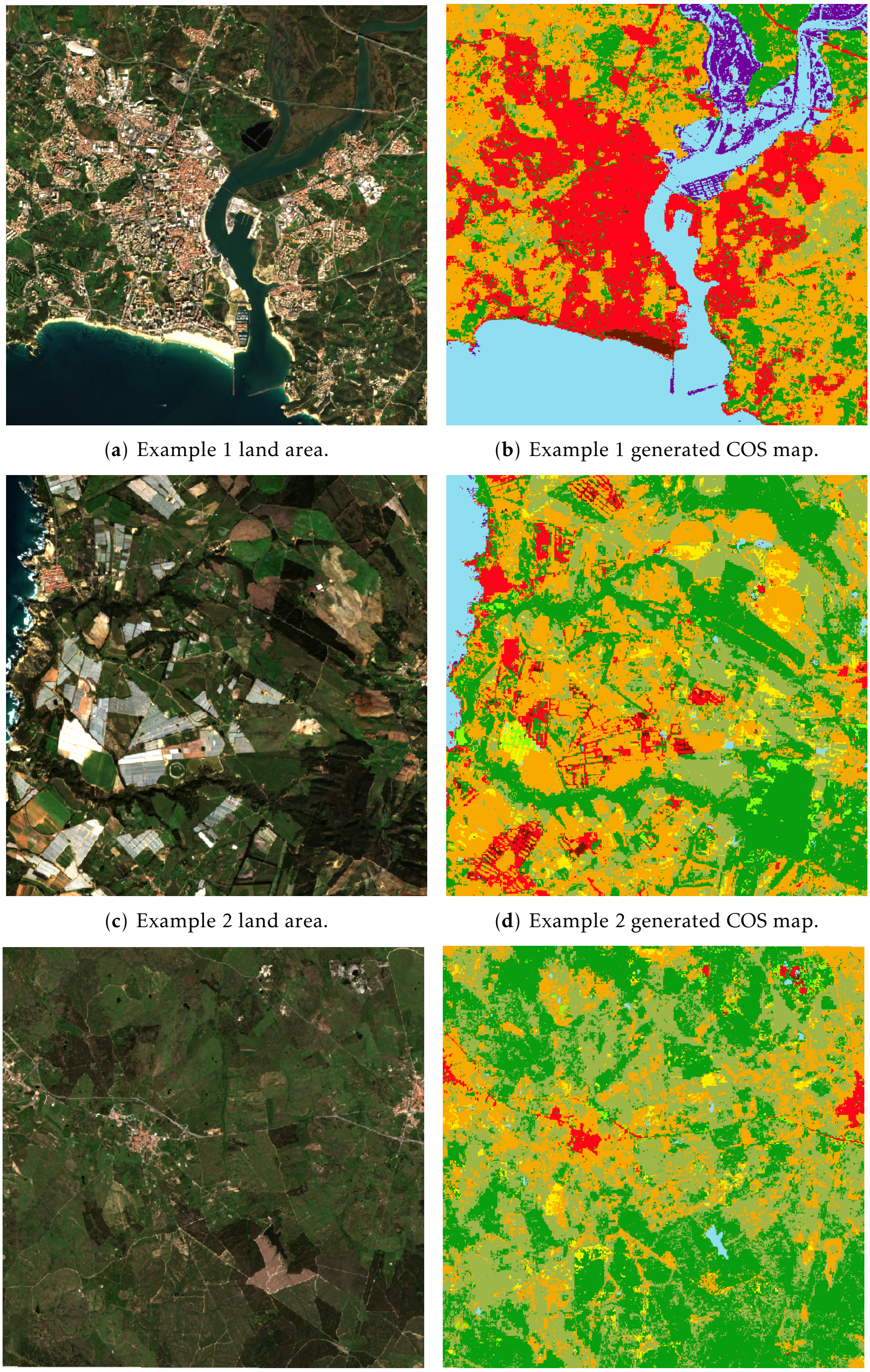
(b) Generated COS map.

(c) Traditional COS map.

Figure 4.14: Direct comparison between traditional simplified and generated COS map.

The generated COS map achieved an accuracy of 53.8% in the specific land area shown in figure 4.14a, however, as previously stated, differences in classification approach between both maps may reveal an accuracy number that is pessimistic and may not be representative of the realistic (i.e. physical ground truth) accuracy.

In conclusion, follows plug-in created examples of Portugal, displayed in figure 4.15.



(**a**) Example 1 land area.

(**b**) Example 1 generated COS map.

(**c**) Example 2 land area.

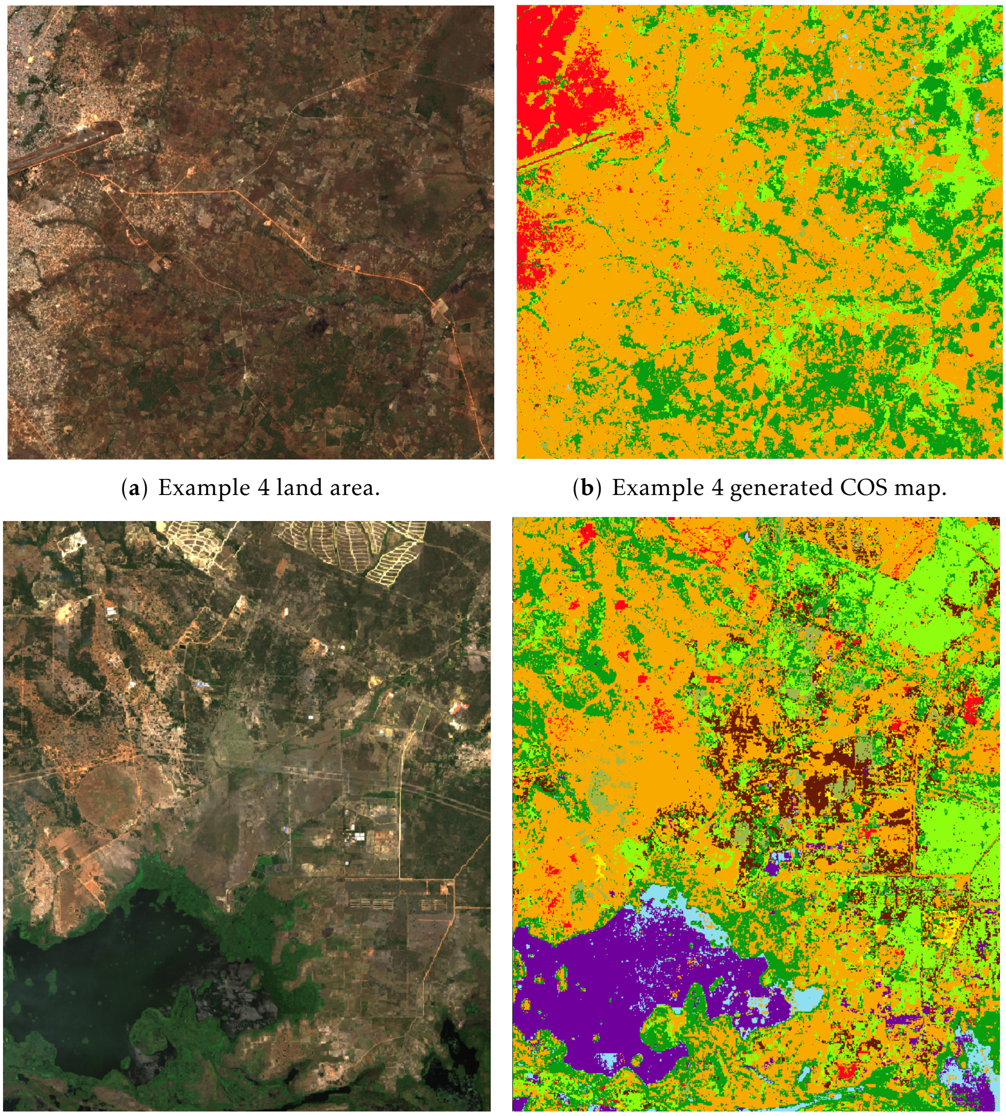
(**d**) Example 2 generated COS map.

(**e**) Example 3 land area.

(**f**) Example 3 generated COS map.

Figure 4.15: Plug-in created Portugal examples.

Given the unrestricted working area of the plug-in, the application can be utilised, with varying degrees of accuracy, in other countries. As an example, follows additional plug-in created examples of Angola, displayed in figure 4.16.



(**a**) Example 4 land area.

(**b**) Example 4 generated COS map.

(**c**) Example 5 land area.

(**d**) Example 5 generated COS map.

Figure 4.16: Plug-in created Angola examples.

Chapter 5

Conclusion

Over the last two decades, machine learning algorithms have been at the focus of land use and land cover applications, benefiting yearly from advances in algorithm optimization, new methods, hardware upgrades, software refining and additional data sources. Each year multiple papers are published world-wide on the topic, comprising of different techniques, distinct satellite data sources and diverse data fusion procedures. The present dissertation focused on implementing an automatic LCLU classification procedure on a widely used geographic information system using modern remote sensing satellite imagery. The project, with Portugal as its focus, aimed to test, integrate and build on previous approaches by utilizing temporal data as additional input variables in order to harvest the contextual information contained in the vegetation cycles.

Throughout the algorithm selection process, three commonly used machine learning algorithms were tested in identical environments along with the progressive integration of multiple techniques in both single and multi temporal approaches. The training procedure, submitted in chapter 4, highlighted the initial unsuitability of every tested algorithm for the required application magnitude when only one season of data was provided. The subsequent addition of vegetation indices also revealed underwhelming results in light of the proclaimed improvements described in the reviewed literature, displaying limited impact in the single temporal implementation. The implementation of the multi temporal approach, on the other hand, revealed significant improvements in the classification of flora-based classes across all trained algorithms, as well as the amplification of the effects of vegetation indices. In turn, validating the multi temporal technique as the optimal approach in the automatic creation of LCLU maps.

The choice of the best algorithm for the intended implementation was critical to the application's success. Among the three tested algorithms, the Support vector machine-based classifier, despite demonstrating the highest affinity for vegetation indices, consistently produced the lowest scores across the testing procedure. Ultimately, the extremely slow training procedure derived from the quadratic increase in training time with the amount of input data render the SVM classifier method unfit for the required implementation's constraints. The Random forest-based classifier achieved the best results in the single temporal approach and second highest in the multi temporal approach, the addition of vegetation indices, however, proved to be detrimental to the classifiers performance. In the end, while the acquired results validate the use of RF classifiers in LCLU map creation, the excessive storage requirements of the algorithm render it as unfit for the intended application. Lastly, the Artificial neural network-based classifier achieved the best results in the multi-temporal approach, while also demonstrating adequate utilization of vegetation indices. In light of the algorithm's speed, scalability and low storage foot print, the ANN classifier was selected as the optimal algorithm for the application.

By exploring the implementation and impact of commonly used techniques and algorithms in a multi temporal approach, the completion of the present dissertation enabled the creation of a proof-of-concept lightweight QGIS plug-in capable of acquiring, processing and automatically classifying any land region of Portugal and, to varying degrees of accuracy, other countries. In turn, achieving the dissertation's goal, but not without its shortcomings. The acquired nine-class out of sight overall accuracy of 65,9% is not yet within the interval of usability, numerous issues still prevent the application from fully replacing the traditional method. The comparable values achieved by all the algorithms, however, appear to indicate that the employed ML method is not the bottleneck of the implementation, rather, the ground truth data used to train the algorithm is plagued with parasitic errors that prevent clearer distinction between the targeted classes. As such, future work in the field will undoubtedly benefit from the increase in the abundance of overall satellite data and updated traditional land cover and land use maps.

5.1 Future work

Despite the positive development of a proof-of-concept automatic LCLU plugin, the implementation still possesses many areas that could be improved and developed further. In the following section, some ideas for future implementations will be introduced along with the targeted challenge.

- **Fusion of additional data sources:** In addition to multi-spectral data, topographic data such as LiDAR might help ease the classification hurdle of inclined surfaces where the employed method occasionally miss classifies the lack of reflection created by scattered light from overlapping foliage as water. Furthermore, the addition of relative land height might help differentiate between the multiple flora-based classes and help analyse water levels.

- **Distinct algorithm architectures:** In opposition to the chosen sub type of artificial neural network, further work could employ the additional temporal data by means of "memory"nodes present in deep learning methods, akin to speech recognition algorithms. Further implementations could also explore the use of Encoder-Decoder recurrent neural network models in order to predict future seasons based on cyclical data.

- **Distinct classification approach:** In contrast with the per-pixel approach pursued in the present dissertation, an object based classification could be explored and the fractal dimension (i.e. roughness) of each drawn object fed as an input variable. The aforementioned approach, despite losing granularity, might help differentiate anthropogenic structures such as crop fields from other vegetation filed areas.

- **Updated per-pixel traditional maps:** As previously noted, the use of adequate ground truth maps is of paramount importance to the general performance of any classifier. Future implementation would benefit greatly from updated and rigorous per-pixel LCLU maps. Such work could initially be accomplished using average automatic classifiers and subsequently refined with human intervention.

Bibliography

[1] M. K. Arora and S. Mathur. "Multi-source classification using artificial neural network in a rugged terrain." In: *Geocarto International* 16.3 (2001), pp. 37–44. ISSN: 10106049. DOI: 10.1080/10106040108542202.

[2] G. F. Hepner, T. Logan, N. Ritter, and N. Bryant. *Artificial neural network classification using a minimal training set: comparison to conventional supervised classification*. Vol. 56. 4. 1990, pp. 469–473.

[3] R. Lawrence, A. Bunn, S. Powell, and M. Zambon. "Classification of remotely sensed imagery using stochastic gradient boosting as a refinement of classification tree analysis." In: *Remote Sensing of Environment* 90.3 (2004), pp. 331–336. ISSN: 00344257. DOI: 10.1016/j.rse.2004.01.007.

[4] A. Schneider. "Monitoring land cover change in urban and peri-urban areas using dense time stacks of Landsat satellite data and a data mining approach." In: *Remote Sensing of Environment* 124 (2012), pp. 689–704. ISSN: 00344257. DOI: 10.1016/j.rse.2012.06.006.

[5] H. chien Shih, D. A. Stow, and Y. H. Tsai. "Guidance on and comparison of machine learning classifiers for Landsat-based land cover and land use mapping." In: *International Journal of Remote Sensing* 40.4 (2019), pp. 1248–1274. ISSN: 13665901. DOI: 10.1080/01431161.2018.1524179. URL: https://www.tandfonline.com/doi/full/10.1080/01431161.2018.1524179.

[6] G. Mountrakis, J. Im, and C. Ogole. *Support vector machines in remote sensing: A review*. 2011. DOI: 10.1016/j.isprsjprs.2010.11.001.

[7] P. H. Verburg, K. Neumann, and L. Nol. *Challenges in using land use and land cover data for global change studies*. 2011. DOI: 10.1111/j.1365-2486.2010.02307.x.

[8] K. J. Wessels, B. Reyers, A. S. Van Jaarsveld, and M. C. Rutherford. "Identification of potential conflict areas between land transformation and biodiversity conservation in north-eastern South Africa." In: *Agriculture, Ecosystems and Environment* 95.1 (2003), pp. 157–178. ISSN: 01678809. DOI: 10.1016/S0167-8809(02)00102-0.

[9] S. Gebhardt, T. Wehrmann, M. A. M. Ruiz, P. Maeda, J. Bishop, M. Schramm, R. Kopeinig, O. Cartus, J. Kellndorfer, R. Ressl, L. A. Santos, and M. Schmidt. "MAD-MEX: Automatic wall-to-wall land cover monitoring for the mexican REDD-MRV program using all landsat data." In: *Remote Sensing* 6.5 (2014), pp. 3923–3943. ISSN: 20724292. DOI: 10.3390/rs6053923. URL: http://www.mdpi.com/2072-4292/6/5/3923.

[10] J Aerts. "Adaptation for river basins: connecting policy goals to the water resources system." In: *Water science and technology : a journal of the International Association on Water Pollution Research* 51.5 (2005), pp. 121–31. ISSN: 0273-1223. URL: http://www.ncbi.nlm.nih.gov/pubmed/15918365.

[11] C. J. Schulp, G. J. Nabuurs, P. H. Verburg, and R. W. de Waal. "Effect of tree species on carbon stocks in forest floor and mineral soil and implications for soil carbon inventories." In: *Forest Ecology and Management* 256.3 (2008), pp. 482–490. ISSN: 03781127. DOI: 10.1016/j.foreco.2008.05.007.

[12] S. Griffin, J. Rogan, and D. Runfola. "Application of spectral and environmental variables to map the Kissimmee Prairie ecosystem using classification trees." In: *GIScience and Remote Sensing* 48.3 (2011), pp. 299–323. ISSN: 15481603. DOI: 10.2747/1548-1603.48.3.299.

[13] B. M. Meneses, E. Reis, M. J. Vale, and R. Reis. "Modelling land use and land cover changes in Portugal: A multi-scale and multi-temporal approach." In: *Finisterra* 53.107 (2018), pp. 3–26. ISSN: 04305027. DOI: 10.18055/finis12258.

[14] *Portugal land cover country fact sheet 2012 — European Environment Agency*. URL: https://www.eea.europa.eu/themes/landuse/land-cover-country-fact-sheets/pt-portugal-landcover-2012.pdf/view (visited on 01/29/2020).

[15] P. Cabral and A. Zamyatin. "Markov processes in modeling land use and land cover changes in Sintra-Cascais, Portugal." In: *DYNA (Colombia)* 76.158 (2009), pp. 191–198. ISSN: 0012-7353. URL: https://novaresearch.unl.pt/en/publications/markov-processes-in-modeling-land-use-and-land-cover-changes-in-s-2.

[16] H. Costa. "Estratégia Multi-temporal para Produção Automática de Cartografia de Ocupação do Solo com Imagens AWiFS." In: (2008), p. 106. URL: http://run.unl.pt/bitstream/10362/2347/1/TSIG0049.pdf.

[17] *FCT — Projectos de I&D — Consultas à base de dados*. 2015. URL: https://www.fct.pt/apoios/projectos/consulta/vglobal_projecto.phtml.pt?idProjecto=154523&idElemConcurso=12344 (visited on 06/22/2021).

[18] *CA3 - Computational Intelligence Research Group*. URL: https://www.ca3-uninova.org/project_ipsters (visited on 06/21/2021).

[19] *IPSentinel*. URL: https://ipsentinel.pt/ (visited on 06/22/2021).

[20] A. L. Samuel. "Some studies in machine learning using the game of checkers." In: *IBM Journal of Research and Development* 44.1-2 (2000), pp. 207–219. ISSN: 00188646. DOI: 10.1147/rd.441.0206. URL: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.368.2254.

[21] P. Langley. *The changing science of machine learning*. 2011. DOI: 10.1007/s10994-011-5242-y.

[22] J. September. *ACTA TECHNICA CORVINIENSIS-Bulletin of Engineering ORGANIZATION AND CHARACTERISTICS OF BUSINESS ZONES*. Tech. rep. URL: https://www.pwc.com/gx/en/industries/industries-www.hypertextbook.com/facts/1999/katrinajones.s.

[23] Y. Lecun, Y. Bengio, and G. Hinton. *Deep learning*. 2015. DOI: 10.1038/nature14539.

[24] Matlab. *Unsupervised Learning - MATLAB & Simulink*. 2016. URL: https://www.mathworks.com/discovery/unsupervised-learning.html.

[25] J. Keuchel, S. Naumann, M. Heiler, and A. Siegmund. "Automatic land cover analysis for Tenerife by supervised classification using remotely sensed data." In: *Remote Sensing of Environment* 86.4 (2003), pp. 530–541. ISSN: 00344257. DOI: 10.1016/S0034-4257(03)00130-5.

[26] A. Mukhopadhyay and U. Maulik. "Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with SVM classifier." In: *IEEE Transactions on Geoscience and Remote Sensing* 47.4 (2009), pp. 1132–1138. ISSN: 01962892. DOI: 10.1109/TGRS.2008.2008182.

[27] A. Gupta. *ML | Semi-Supervised Learning - GeeksforGeeks*. 2018. URL: https://www.geeksforgeeks.org/ml-semi-supervised-learning/ (visited on 01/30/2020).

[28] R. R. Vatsavai, S. Shekhar, and T. E. Burk. "A semi-supervised learning method for remote sensing data mining." In: *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*. Vol. 2005. 2005, pp. 207–211. ISBN: 0769524885. DOI: 10.1109/ICTAI.2005.17.

[29] D. Tuia and G. Camps-Valls. "Semisupervised remote sensing image classification with cluster kernels." In: *IEEE Geoscience and Remote Sensing Letters* 6.2 (2009), pp. 224–228. ISSN: 1545598X. DOI: 10.1109/LGRS.2008.2010275. URL: http://ieeexplore.ieee.org..

[30] R. G. Congalton. "A review of assessing the accuracy of classifications of remotely sensed data." In: *Remote Sensing of Environment* 37.1 (1991), pp. 35–46. ISSN: 00344257. DOI: 10.1016/0034-4257(91)90048-B.

[31] R. R. Colditz. "An evaluation of different training sample allocation schemes for discrete and continuous land cover classification using decision tree-based algorithms." In: *Remote Sensing* 7.8 (2015), pp. 9655–9681. ISSN: 20724292. DOI: 10.3390/rs70809655. URL: http://www.mdpi.com/2072-4292/7/8/9655.

[32] P. Thanh Noi and M. Kappas. "Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery." In: *Sensors (Basel, Switzerland)* 18.1 (2017). ISSN: 14248220. DOI: 10.3390/s18010018.

[33] D. Lu and Q. Weng. *A survey of image classification methods and techniques for improving classification performance*. 2007. DOI: 10.1080/01431160600746456.

[34] K. Duan, S. S. Keerthi, and A. N. Poo. "Evaluation of simple performance measures for tuning SVM hyperparameters." In: *Neurocomputing* 51 (2003), pp. 41–59. ISSN: 09252312. DOI: 10.1016/S0925-2312(02)00601-X.

[35] É. M. Pôssa and P. Maillard. "Precise Delineation of Small Water Bodies from Sentinel-1 Data using Support Vector Machine Classification." In: *Canadian Journal of Remote Sensing* 44.3 (2018), pp. 179–190. ISSN: 17127971. DOI: 10.1080/07038992.2018.1478723. URL: https://www.tandfonline.com/doi/full/10.1080/07038992.2018.1478723.

[36] J. R. Jensen and K. Lulla. "Introductory digital image processing: A remote sensing perspective." In: *Geocarto International* 2.1 (1987), p. 65. ISSN: 10106049. DOI: 10.1080/10106048709354084.

[37] S. S. Heydari and G. Mountrakis. "Effect of classifier selection, reference sample size, reference class distribution and scene heterogeneity in per-pixel classification accuracy using 26 Landsat sites." In: *Remote Sensing of Environment* 204 (2018), pp. 648–658. ISSN: 00344257. DOI: 10.1016/j.rse.2017.09.035.

[38] J. W. Strijbos, R. L. Martens, F. J. Prins, and W. M. Jochems. "Content analysis: What are they talking about?" In: *Computers and Education* 46.1 (2006), pp. 29–48. ISSN: 03601315. DOI: 10.1016/j.compedu.2005.04.002. URL: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.397.5780.

[39] H. He and E. A. Garcia. "Learning from imbalanced data." In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), pp. 1263–1284. ISSN: 10414347. DOI: 10.1109/TKDE.2008.239.

[40] *Registo Nacional de Dados Geográficos - Direção-Geral do Território*. 2018. URL: https://snig.dgterritorio.gov.pt/rndg/srv/por/catalog.search#/metadata/b498e89c-1093-4793-ad22-63516062891b (visited on 10/03/2021).

[41] C. Pohl and J. L. Van Genderen. *Review article Multisensor image fusion in remote sensing: Concepts, methods and applications*. 1998. DOI: 10.1080/014311698215748.

[42] J. Zhang. *No Title*. 2010. DOI: 10.1080/19479830903561035. URL: http://www.tandfonline.com/doi/abs/10.1080/19479830903561035.

[43] A. H. S. Solberg. *Data fusion for remote sensing applications*. Tech. rep. 2006.

[44] L. R. Mansaray, F. Wang, J. Huang, L. Yang, and A. S. Kanu. "Accuracies of support vector machine and random forest in rice mapping with Sentinel-1A, Landsat-8 and Sentinel-2A datasets." In: *Geocarto International* (2019), pp. 1–21. ISSN: 1010-6049. DOI: 10.1080/10106049.2019.1568586. URL: https://www.tandfonline.com/doi/full/10.1080/10106049.2019.1568586.

[45] A. O. Onojeghuo, G. A. Blackburn, Q. Wang, P. M. Atkinson, D. Kindred, and Y. Miao. "No Title." In: *International Journal of Remote Sensing* 39.4 (2018), pp. 1042–1067. DOI: 10.1080/01431161.2017.1395969. URL: https://www.tandfonline.com/doi/full/10.1080/01431161.2017.1395969.

[46] X. Song, Z. Duan, and X. Jiang. "Comparison of artificial neural networks and support vector machine classifiers for land cover classification in Northern China using a SPOT-5 HRG image." In: *International Journal of Remote Sensing* 33.10 (2012), pp. 3301–3320. ISSN: 13665901. DOI: 10.1080/01431161.2011.568531. URL: https://www.tandfonline.com/doi/full/10.1080/01431161.2011.568531.

[47] A. Ghosh and P. K. Joshi. "A comparison of selected classification algorithms for mappingbamboo patches in lower Gangetic plains using very high resolution WorldView 2 imagery." In: *International Journal of Applied Earth Observation and Geoinformation* 26.1 (2014), pp. 298–311. ISSN: 15698432. DOI: 10.1016/j.jag.2013.08.011.

[48] E. Raczko and B. Zagajewski. "Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral APEX images." In: *European Journal of Remote Sensing* 50.1 (2017), pp. 144–154. ISSN: 22797254. DOI: 10.1080/22797254.2017.1299557. URL: https://www.tandfonline.com/doi/full/10.1080/22797254.2017.1299557.

[49] Q. Yu, P. Gong, N. Clinton, G. Biging, M. Kelly, and D. Schirokauer. "Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery." In: *Photogrammetric Engineering and Remote Sensing* 72.7 (2006), pp. 799–811. ISSN: 00991112. DOI: 10.14358/PERS.72.7.799.

[50] C. Sothe, C. M. De Almeida, M. B. Schimalski, V. Liesenberg, L. E. La Rosa, J. D. Castro, and R. Q. Feitosa. "A comparison of machine and deep-learning algorithms applied to multisource data for a subtropical forest area classification." In: *International Journal of Remote Sensing* 41.5 (2020), pp. 1943–1969. ISSN: 13665901. DOI: 10.1080/01431161.2019.1681600. URL: https://www.tandfonline.com/doi/full/10.1080/01431161.2019.1681600.

[51] B. Fu, Y. Wang, A. Campbell, Y. Li, B. Zhang, S. Yin, Z. Xing, and X. Jin. "Comparison of object-based and pixel-based Random Forest algorithm for wetland vegetation mapping using high spatial resolution GF-1 and SAR data." In: *Ecological Indicators* 73 (2017), pp. 105–117. ISSN: 1470160X. DOI: 10.1016/j.ecolind.2016.09.029.

[52] X. Guan, S. Liao, J. Bai, F. Wang, Z. Li, Q. Wen, J. He, and T. Chen. "Urban land-use classification by combining high-resolution optical and long-wave infrared images." In: *Geo-Spatial Information Science* 20.4 (2017), pp. 299–308. ISSN: 10095020. DOI: 10.1080/10095020.2017.1403731.

[53] A. M. Demers, S. N. Banks, J. Pasher, J. Duffe, and S. Laforest. "A comparative analysis of object-based and pixel-based classification of RADARSAT-2 C-band and optical satellite data for mapping shoreline types in the canadian arctic." In: *Canadian Journal of Remote Sensing* 41.1 (2015), pp. 1–19. ISSN: 17127971. DOI: 10.1080/07038992.2015.1020361.

[54] L. Dingle Robertson, D. J. King, and C. Davies. "Object-based image analysis of optical and radar variables for wetland evaluation." In: *International Journal of Remote Sensing* 36.23 (2015), pp. 5811–5841. ISSN: 13665901. DOI: 10.1080/01431161.2015.1109727. URL: https://www.tandfonline.com/doi/full/10.1080/01431161.2015.1109727.

[55] N. J. Lantz and J. Wang. "Object-based classification of Worldview-2 imagery for mapping invasive common reed, Phragmites Australis." In: *Canadian Journal of Remote Sensing* 39.4 (2013), pp. 328–340. ISSN: 07038992. DOI: 10.5589/m13-041.

[56] T. Blaschke. *Object based image analysis for remote sensing*. 2010. DOI: 10.1016/j.isprsjprs.2009.06.004.

[57] T. Warner. "Kernel-Based Texture in Remote Sensing Image Classification." In: *Geography Compass* 5.10 (2011), pp. 781–798. ISSN: 17498198. DOI: 10.1111/j.1749-8198.2011.00451.x. URL: http://doi.wiley.com/10.1111/j.1749-8198.2011.00451.x.

[58] G. P. Asner. "Automated mapping of tropical deforestation and forest degradation: CLASlite." In: *Journal of Applied Remote Sensing* 3.1 (2009), p. 033543. ISSN: 1931-3195. DOI: 10.1117/1.3223675. URL: http://remotesensing.spiedigitallibrary.org/article.aspx?doi=10.1117/1.3223675.

[59] J. Irons, M. Taylor, and L. Rocchio. *Landsat Science*. 2013. URL: https://landsat.gsfc.nasa.gov/landsat-1/http://landsat.gsfc.nasa.gov/?page{_}id=5377http://landsat.gsfc.nasa.gov/?page{_}id=5377 (visited on 02/01/2020).

[60] N. Joshi, M. Baumann, A. Ehammer, R. Fensholt, K. Grogan, P. Hostert, M. R. Jepsen, T. Kuemmerle, P. Meyfroidt, E. T. Mitchard, J. Reiche, C. M. Ryan, and B. Waske. *A review of the application of optical and radar remote sensing data fusion to land use mapping and monitoring*. 2016. DOI: 10.3390/rs8010070. URL: http://www.mdpi.com/2072-4292/8/1/70.

[61] ESA. *Sentinel Online*. 2019. URL: https://sentinel.esa.int/web/sentinel/missions/sentinel-2/overviewhttps://sentinel.esa.int/web/sentinel/missions/sentinel-2{\%}0Ahttps://sentinel.esa.int/web/sentinel/missions (visited on 02/01/2020).

[62] F. Y. Cheng and D. W. Byun. "Application of high resolution land use and land cover data for atmospheric modeling in the Houston-Galveston metropolitan area, Part I: Meteorological simulation results." In: *Atmospheric Environment* 42.33 (2008), pp. 7795–7811. ISSN: 13522310. DOI: 10.1016/j.atmosenv.2008.04.055.

[63] S. Berberoglu and A. Akin. "Assessing different remote sensing techniques to detect land use/cover changes in the eastern Mediterranean." In: *International Journal of Applied Earth Observation and Geoinformation* 11.1 (2009), pp. 46–53. ISSN: 15698432. DOI: 10.1016/j.jag.2008.06.002.

[64] L. Ji, L. Zhang, and B. Wylie. "Analysis of dynamic thresholds for the normalized difference water index." In: *Photogrammetric Engineering and Remote Sensing* 75.11 (2009), pp. 1307–1317. ISSN: 00991112. DOI: 10.14358/PERS.75.11.1307.

[65] S. K. Jain, R. D. Singh, M. K. Jain, and A. K. Lohani. "Delineation of flood-prone areas using remote sensing techniques." In: *Water Resources Management* 19.4 (2005), pp. 333–347. ISSN: 09204741. DOI: 10.1007/s11269-005-3281-5.

[66] K. Segl, S. Roessner, U. Heiden, and H. Kaufmann. "Fusion of spectral and shape features for identification of urban surface cover types using reflective and thermal hyperspectral data." In: *ISPRS Journal of Photogrammetry and Remote Sensing*. Vol. 58. 1-2. Elsevier, 2003, pp. 99–112. DOI: 10.1016/S0924-2716(03)00020-0.

[67] H. D. Armono, B. G. Mahaputra, and M. Zikra. "The Usage of Geographical Information System in the Selection of Floating Cages Location for Aquaculture at Prigi Bay, Trenggalek Regency, East Java." In: *IOP Conference Series: Earth and Environmental Science*. Vol. 135. 1. Institute of Physics Publishing, 2018. DOI: 10.1088/1755-1315/135/1/012023.

[68] J. Wang, X. Li, X. Sun, and Q. Liu. "Component temperatures inversion for remote sensing pixel based on directional thermal radiation model." In: *Science in China, Series E: Technological Sciences* 43.SUPPL. (2000), pp. 46–47. ISSN: 10069321. DOI: 10.1007/bf02916577.

[69] C. Zhang and Z. Xie. "Data fusion and classifier ensemble techniques for vegetation mapping in the coastal Everglades." In: *Geocarto International* 29.3 (2014), pp. 228–243. ISSN: 10106049. DOI: 10.1080/10106049.2012.756940. URL: http://www.tandfonline.com/doi/abs/10.1080/10106049.2012.756940.

[70] X. Huang and L. Zhang. "A comparative study of spatial approaches for urban mapping using hyperspectral ROSIS images over Pavia City, northern Italy." In: *International Journal of Remote Sensing* 30.12 (2009), pp. 3205–3221. ISSN: 13665901. DOI: 10.1080/01431160802559046.

[71] F. Pacifici, M. Chini, and W. J. Emery. "A neural network approach using multi-scale textural metrics from very high-resolution panchromatic imagery for urban land-use classification." In: *Remote Sensing of Environment* 113.6 (2009), pp. 1276–1292. ISSN: 00344257. DOI: 10.1016/j.rse.2009.02.014.

[72] X. Huang, L. Zhang, and P. Li. "A multiscale feature fusion approach for classification of very high resolution satellite imagery based on wavelet transform." In: *International Journal of Remote Sensing* 29.20 (2008), pp. 5923–5941. ISSN: 13665901. DOI: 10.1080/01431160802139922.

[73] B. Brisco, N. Short, J. Van Der Sanden, R. Landry, and D. Raymond. "A semi-automated tool for surface water mapping with RADARSAT-1." In: *Canadian Journal of Remote Sensing* 35.4 (2009), pp. 336–344. ISSN: 17127971. DOI: 10.5589/m09-025.

[74] M. Imangholiloo, J. Rasinmäki, Y. Rauste, and M. Holopainen. *Utilizing Sentinel-1A Radar Images for Large-Area Land Cover Mapping with Machine-learning Methods*. 2019. DOI: 10.1080/07038992.2019.1635877. URL: https://www.tandfonline.com/doi/full/10.1080/07038992.2019.1635877.

[75] NASA. *Remote Sensors | Earthdata*. 2018. URL: https://earthdata.nasa.gov/learn/remote-sensors{\#}hyperspectralhttps://earthdata.nasa.gov/user-resources/remote-sensors (visited on 02/01/2020).

[76] T. S. W. Marshall Honorof. *What is Lidar — Lidar vs.* 2013. URL: http://www.technewsdaily.com/18375-what-is-lidar.html (visited on 02/01/2020).

[77] W. Y. Yan, A. Shaker, and N. El-Ashmawy. *Urban land cover classification using airborne LiDAR data: A review*. 2015. DOI: 10.1016/j.rse.2014.11.001.

[78] K. D. Fieber, I. J. Davenport, J. M. Ferryman, R. J. Gurney, J. P. Walker, and J. M. Hacker. "Analysis of full-waveform LiDAR data for classification of an orange orchard scene." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 82 (2013), pp. 63–82. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2013.05.002.

[79] L. Du, X. You, K. Li, L. Meng, G. Cheng, L. Xiong, and G. Wang. "Multi-modal deep learning for landform recognition." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 158 (2019), pp. 63–75. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2019.09.018.

[80] C. A. Baldeck, G. P. Asner, R. E. Martin, C. B. Anderson, D. E. Knapp, J. R. Kellner, and S. J. Wright. "Operational tree species mapping in a diverse tropical forest with airborne imaging spectroscopy." In: *PLoS ONE* 10.7 (2015). Ed. by L. Kumar, e0118403. ISSN: 19326203. DOI: 10.1371/journal.pone.0118403. URL: https://dx.plos.org/10.1371/journal.pone.0118403.

[81] L. Zhang, L. Zhang, and B. Du. "Deep learning for remote sensing data: A technical tutorial on the state of the art." In: *IEEE Geoscience and Remote Sensing Magazine* 4.2 (2016), pp. 22–40. ISSN: 21686831. DOI: 10.1109/MGRS.2016.2540798.

[82] F. Agüera, F. J. Aguilar, and M. A. Aguilar. "Using texture analysis to improve per-pixel classification of very high resolution images for mapping plastic greenhouses." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 63.6 (2008), pp. 635–646. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2008.03.003.

[83] B. Ghimire, J. Rogan, and J. Miller. "Contextual land-cover classification: Incorporating spatial dependence in land-cover classification models using random forests and the Getis statistic." In: *Remote Sensing Letters* 1.1 (2010), pp. 45–54. ISSN: 2150704X. DOI: 10.1080/01431160903252327.

[84] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez. "An assessment of the effectiveness of a random forest classifier for land-cover classification." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 67.1 (2012), pp. 93–104. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2011.11.002.

[85] J. Haas and Y. Ban. "Sentinel-1A SAR and sentinel-2A MSI data fusion for urban ecosystem service mapping." In: *Remote Sensing Applications: Society and Environment* 8.6 (2017), pp. 41–53. ISSN: 23529385. DOI: 10.1016/j.rsase.2017.07.006. URL: http://www.mdpi.com/2072-4292/9/6/596.

[86] M. Pesaresi, C. Corbane, A. Julea, A. J. Florczyk, V. Syrris, and P. Soille. "Assessment of the added-value of sentinel-2 for detecting built-up areas." In: *Remote Sensing* 8.4 (2016), p. 299. ISSN: 20724292. DOI: 10.3390/rs8040299. URL: http://www.mdpi.com/2072-4292/8/4/299.

[87] European Spatial Agency (ESA). *MSI Instrument – Sentinel-2 MSI Technical Guide – Sentinel Online*. 2019. URL: https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrumenthttps://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument (visited on 02/02/2020).

[88] J. Gray. "Getting Started With Quantum GIS." In: *Linux Journal* (2008), pp. 1–7. URL: https://www.linuxjournal.com/content/getting-started-quantum-gishttp://www.linuxjournal.com/content/getting-started-quantum-gis.

[89] A. E. Maxwell, T. A. Warner, and F. Fang. *Implementation of machine-learning classification in remote sensing: An applied review*. 2018. DOI: 10.1080/01431161.2018.1433343.

[90] *Daily chart - Python is becoming the world's most popular coding language | Graphic detail | The Economist*. URL: https://www.economist.com/graphic-detail/2018/07/26/python-is-becoming-the-worlds-most-popular-coding-language (visited on 02/04/2020).

[91] Python Foundation. *About Python™ | Python.org*. 2016. URL: https://www.python.org/about/ (visited on 02/04/2020).

[92] Wes McKinney. "Data Structures for Statistical Computing in Python." In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56 –61. DOI: 10.25080/Majora-92bf1922-00a.

[93] T. pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[94] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. "Array programming with NumPy." In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[95] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[96] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. "API design for machine learning software: experiences from the scikit-learn project." In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[97] F. Chollet et al. *Keras*. https://keras.io. 2015.

[98] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[99] J. D. Hunter. “Matplotlib: A 2D graphics environment.” In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[100] R. Al-Rfou et al. “Theano: A Python framework for fast computation of mathematical expressions.” In: *arXiv e-prints* abs/1605.02688 (May 2016). URL: http://arxiv.org/abs/1605.02688.

[101] M. Thompson, R. O. Duda, and P. E. Hart. “Pattern Classification and Scene Analysis.” In: *Leonardo* 7.4 (1974), p. 370. ISSN: 0024094X. DOI: 10.2307/1573081.

[102] H. Franco-Lopez, A. R. Ek, and M. E. Bauer. “Estimation and mapping of forest stand density, volume, and cover type using the k-nearest neighbors method.” In: *Remote Sensing of Environment* 77.3 (2001), pp. 251–274. ISSN: 00344257. DOI: 10.1016/S0034-4257(01)00209-7.

[103] D. Guidici and M. L. Clark. “One-dimensional convolutional neural network land-cover classification of multi-seasonal hyperspectral imagery in the San Francisco Bay Area, California.” In: *Remote Sensing* 9.6 (2017), p. 629. ISSN: 20724292. DOI: 10.3390/rs9060629. URL: http://www.mdpi.com/2072-4292/9/6/629.

[104] V. VAPNIK. “Pattern recognition using generalized portrait method.” In: *Automation and Remote Control* 24 (1963), pp. 774–780.

[105] G. M. Foody and A. Mathur. “Toward intelligent training of supervised image classifications: Directing training data acquisition for SVM classification.” In: *Remote Sensing of Environment* 93.1-2 (2004), pp. 107–117. ISSN: 00344257. DOI: 10.1016/j.rse.2004.06.017.

[106] J. Knorn, A. Rabe, V. C. Radeloff, T. Kuemmerle, J. Kozak, and P. Hostert. “Land cover mapping of large areas using chain classification of neighboring Landsat satellite images.” In: *Remote Sensing of Environment* 113.5 (2009), pp. 957–964. ISSN: 00344257. DOI: 10.1016/j.rse.2009.01.010.

[107] L. Ballanti, L. Blesius, E. Hines, and B. Kruse. “Tree species classification using hyperspectral imagery: A comparison of two classifiers.” In: *Remote Sensing* 8.6 (2016), p. 445. ISSN: 20724292. DOI: 10.3390/rs8060445. URL: http://www.mdpi.com/2072-4292/8/6/445.

[108] P. Mantero, G. Moser, and S. B. Serpico. “Partially supervised classification of remote sensing images through SVM-based probability density estimation.” In: *IEEE Transactions on Geoscience and Remote Sensing* 43.3 (2005), pp. 559–570. ISSN: 01962892. DOI: 10.1109/TGRS.2004.842022.

[109] N. T. Son, C. F. Chen, C. R. Chen, and V. Q. Minh. “Assessment of Sentinel-1A data for rice crop classification using random forests and support vector machines.” In: *Geocarto International* 33.6 (2018), pp. 587–601. ISSN: 10106049. DOI: 10.1080/10106049.2017.1289555.

[110] C. Cortes, C. Cortes, and V. Vapnik. “Support-Vector Networks.” In: *MACHINE LEARNING* 20 (1995), pp. 273–297. URL: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.15.9362.

[111] H. Kandan. “Understanding the kernel trick. – Towards Data Science.” In: *Towards Data Science* (2017), pp. 1/1. URL: https://towardsdatascience.com/understanding-the-kernel-trick-e0bc6112ef78.

[112] T. K. Ho. “Random decision forests.” In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*. Vol. 1. IEEE Computer Society, 1995, pp. 278–282. ISBN: 0818671289. DOI: 10.1109/ICDAR.1995.598994.

[113] A. M. Prasad, L. R. Iverson, and A. Liaw. “Newer classification and regression tree techniques: Bagging and random forests for ecological prediction.” In: *Ecosystems* 9.2 (2006), pp. 181–199. ISSN: 14329840. DOI: 10.1007/s10021-005-0054-1.

[114] E. Adam, O. Mutanga, J. Odindi, and E. M. Abdel-Rahman. “Land-use/cover classification in a heterogeneous coastal landscape using RapidEye imagery: evaluating the performance of random forest and support vector machines classifiers.” In: *International Journal of Remote Sensing* 35.10 (2014), pp. 3440–3458. ISSN: 0143-1161. DOI: 10.1080/01431161.2014.903435. URL: https://www.tandfonline.com/doi/full/10.1080/01431161.2014.903435.

[115] R. Bellman. *Dynamic Programming (Reprinted version)*. Dover Publications, 2003, p. 384. ISBN: 0486428095.

[116] N. T. Son, C. F. Chen, C. R. Chen, and V. Q. Minh. “Assessment of Sentinel-1A data for rice crop classification using random forests and support vector machines.” In: *Geocarto International* 33.6 (2018), pp. 587–601. ISSN: 10106049. DOI: 10.1080/10106049.2017.1289555. URL: https://www.tandfonline.com/doi/full/10.1080/10106049.2017.1289555.

[117] Z. Wang, C. Lai, X. Chen, B. Yang, S. Zhao, and X. Bai. “Flood hazard risk assessment model based on random forest.” In: *Journal of Hydrology* 527 (2015), pp. 1130–1141. ISSN: 00221694. DOI: 10.1016/j.jhydrol.2015.06.008.

[118] L. Breiman. “Random forests.” In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 08856125. DOI: 10.1023/A:1010933404324.

[119] D. C. Duro, S. E. Franklin, and M. G. Dubé. “A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery.” In: *Remote Sensing of Environment* 118 (2012), pp. 259–272. ISSN: 00344257. DOI: 10.1016/j.rse.2011.11.020.

[120] A. L. Wiener and M. *Classification and Regression by randomForest. R News 2.* 2003. URL: https://www.researchgate.net/publication/228451484.

[121] L. Breiman. “Bagging predictors.” In: *Machine Learning* 24.2 (1996), pp. 123–140. ISSN: 08856125. DOI: 10.1007/bf00058655.

[122] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson. “Random forests for land cover classification.” In: *Pattern Recognition Letters*. Vol. 27. 4. 2006, pp. 294–300. DOI: 10.1016/j.patrec.2005.08.011.

[123] V. F. Rodriguez-Galiano, M. Chica-Olmo, F. Abarca-Hernandez, P. M. Atkinson, and C. Jeganathan. “Random Forest classification of Mediterranean land cover using multi-seasonal imagery and multi-seasonal texture.” In: *Remote Sensing of Environment* 121 (2012), pp. 93–107. ISSN: 00344257. DOI: 10.1016/j.rse.2011.12.003.

[124] T. Kavzoglu and P. M. Mather. “The use of backpropagating artificial neural networks in land cover classification.” In: *International Journal of Remote Sensing* 24.23 (2003), pp. 4907–4938. ISSN: 01431161. DOI: 10.1080/0143116031000114851.

[125] S. J. Kwon. *Artificial neural networks.* Vol. 91. S8. John Wiley & Sons, Ltd, 2011, pp. 1–426. ISBN: 9781617615535. DOI: 10.4324/9781315154282-3.

[126] D. R. Peddle, E. F. Ledrew, G. M. Foody, A. Zhang, and S. E. Franklin. “Multisource image classification II: An empirical comparison of evidential reasoning and neural network approaches.” In: *Canadian Journal of Remote Sensing* 20.4 (1994), pp. 396–407. ISSN: 17127971. DOI: 10.1080/07038992.1994.10874582.

[127] P Gong. “Integrated analysis of spatial data from multiple sources: Using evidential reasoning and artificial neural network techniques for geological mapping.” In: *Photogrammetric Engineering and Remote Sensing* 62.5 (1996), pp. 513–523. ISSN: 00991112.

[128] R. M. De Oliveira, R. C. Araújo, F. J. Barros, A. P. Segundo, R. F. Zampolo, W. Fonseca, V. Dmitriev, and F. S. Brasil. “A system based on artificial neural networks for automatic classification of hydro-generator stator windings partial discharges.” In: *Journal of Microwaves, Optoelectronics and Electromagnetic Applications* 16.3 (2017), pp. 628–645. ISSN: 21791074. DOI: 10.1590/2179-10742017v16i3854.

[129] Y. Li, H. Zhang, and Q. Shen. “Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network.” In: *Remote Sensing* 9.1 (2017), p. 67. ISSN: 20724292. DOI: 10.3390/rs9010067. URL: http://www.mdpi.com/2072-4292/9/1/67.

[130] M. D. Zeiler and R. Fergus. “LNCS 8689 - Visualizing and Understanding Convolutional Networks - eccv2014.pdf.” In: *ECCV*. 2014, pp. 818–833. URL: https://cs.nyu.edu/{~}fergus/papers/zeilerECCV2014.pdf{\%}0Ahttp://www.matthewzeiler.com/pubs/arxive2013/eccv2014.pdf.

[131] C. Farabet, C. Couprie, L. Najman, and Y. Lecun. “Learning hierarchical features for scene labeling.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1915–1929. ISSN: 01628828. DOI: 10.1109/TPAMI.2012.231.

[132] N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. “Deep Neural Networks for Acoustic Modeling in Speech Recognition.” In: *IEEE SIGNAL PROCESSING MAGAZINE* 2.november (2012), pp. 1–27. ISSN: 1053-5888. DOI: 10.1109/MSP.2012.2205597.

[133] T Sainath, A.-R. Mohamed, B Kingsbury, and B Ramabhadran. *Proc. Acoustics, Speech and Signal Processing*. 2013.

[134] T Ciodaro, D Deva, J. M. De Seixas, and D Damazio. “Online particle detection with neural networks based on topological calorimetry information.” In: *Journal of Physics: Conference Series*. Vol. 368. 1. 2012. DOI: 10.1088/1742-6596/368/1/012030.

[135] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey. “Deep learning of the tissue-regulated splicing code.” In: *Bioinformatics* 30.12 (2014). ISSN: 14602059. DOI: 10.1093/bioinformatics/btu277.

[136] R Salakhutdinov and G Hinton. *Proc. International Conference on Artificial Intelligence and Statistics*. 2009.

[137] A. M. Saxe, J. L. McClelland, and S. Ganguli. “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks.” In: *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. 2014. arXiv: 1312.6120. URL: http://arxiv.org/abs/1312.6120.

[138] S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” In: *32nd International Conference on Machine Learning, ICML 2015*. Vol. 1. International Machine Learning Society (IMLS), 2015, pp. 448–456. ISBN: 9781510810587. arXiv: 1502.03167.

[139] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. “Backpropagation Applied to Handwritten Zip Code Recognition.” In: *Neural Computation* 1.4 (1989), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541.

[140] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: A simple way to prevent neural networks from overfitting.” In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. ISSN: 15337928.

[141] R. K. Srivastava, K. Greff, and J. Schmidhuber. “Highway Networks.” In: (2015). arXiv: 1505.00387. URL: http://arxiv.org/abs/1505.00387.

[142] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition.” In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-Decem. IEEE Computer Society, 2016, pp. 770–778. ISBN: 9781467388504. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

[143] K. Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.” In: *Biological Cybernetics* 36.4 (1980), pp. 193–202. ISSN: 03401200. DOI: 10.1007/BF00344251.

[144] C. Huang, L. S. Davis, and J. R. Townshend. “An assessment of support vector machines for land cover classification.” In: *International Journal of Remote Sensing* 23.4 (2002), pp. 725–749. ISSN: 01431161. DOI: 10.1080/01431160110040323.

[145] A. Puissant, S. Rougiera, and A. Stumpf. “Object-oriented mapping of urban trees using random forestclassifiers.” In: *International Journal of Applied Earth Observation and Geoinformation* 26.1 (2014), pp. 235–245. ISSN: 15698432. DOI: 10.1016/j.jag.2013.07.002.

[146] K. Tatsumi, Y. Yamashiki, M. A. Canales Torres, and C. L. R. Taipe. “Crop classification of upland fields using Random forest of time-series Landsat 7 ETM+ data.” In: *Computers and Electronics in Agriculture* 115 (2015), pp. 171–179. ISSN: 01681699. DOI: 10.1016/j.compag.2015.05.001.

[147] M. Belgiu and L. Drăgu. *Random forest in remote sensing: A review of applications and future directions*. 2016. DOI: 10.1016/j.isprsjprs.2016.01.011.

[148] X. Shang and L. A. Chisholm. “Classification of Australian native forest species using hyperspectral remote sensing and machine-learning classification algorithms.” In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.6 (2014), pp. 2481–2489. ISSN: 21511535. DOI: 10.1109/JSTARS.2013.2282166.

[149] H. Han, S. Lee, J. Im, M. Kim, M. I. Lee, M. H. Ahn, and S. R. Chung. "Detection of convective initiation using Meteorological Imager onboard Communication, Ocean, and Meteorological Satellite based on machine learning approaches." In: *Remote Sensing* 7.7 (2015), pp. 9184–9204. ISSN: 20724292. DOI: 10.3390/rs70709184. URL: http://www.mdpi.com/2072-4292/7/7/9184.

[150] J. C. W. Chan and D. Paelinckx. "Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery." In: *Remote Sensing of Environment* 112.6 (2008), pp. 2999–3011. ISSN: 00344257. DOI: 10.1016/j.rse.2008.02.011.

[151] D. Shi and X. Yang. "Support Vector Machines for Land Cover Mapping from Remote Sensor Imagery." In: 2015, pp. 265–279. DOI: 10.1007/978-94-017-9813-6_13. URL: http://link.springer.com/10.1007/978-94-017-9813-6{_}13.

[152] Direção-Geral do Território. *Especificações Técnicas da Carta de Uso e Ocupação de Solo (COS) de Portugal Continental para 2018*. Tech. rep. 2019, pp. 1–60. URL: http://mapas.dgterritorio.pt/atom-dgt/pdf-cous/COS2018/ET-COS-2018_v1.pdf..

[153] D. Solovey and A. Militar. *Terrain Classification from High Resolution Aerial Images Using Deep Learning*. Tech. rep. 2017.

[154] J. Xue and B. Su. *Significant remote sensing vegetation indices: A review of developments and applications*. 2017. DOI: 10.1155/2017/1353691.

[155] O. E. Adeyeri, A. A. Akinsanola, and K. A. Ishola. "Investigating surface urban heat island characteristics over Abuja, Nigeria: Relationship between land surface temperature and multiple vegetation indices." In: *Remote Sensing Applications: Society and Environment* 7 (2017), pp. 57–68. ISSN: 23529385. DOI: 10.1016/j.rsase.2017.06.005.

[156] C. Polykretis, M. G. Grillakis, and D. D. Alexakis. "Exploring the impact of various spectral indices on land cover change detection using change vector analysis: A case study of Crete Island, Greece." In: *Remote Sensing* 12.2 (2020), p. 319. ISSN: 20724292. DOI: 10.3390/rs12020319. URL: https://www.mdpi.com/2072-4292/12/2/319.

[157] P. Piekarski and Z. Zwoliński. "Temporal variation in vegetation indexes for pine and beech stands during the vegetation season, Szczecin Lowland, Poland." In: *Quaestiones Geographicae* 33.3 (2014), pp. 131–143. ISSN: 20816383. DOI: 10.2478/quageo-2014-0037. URL: www.geoportal.gov.pl..

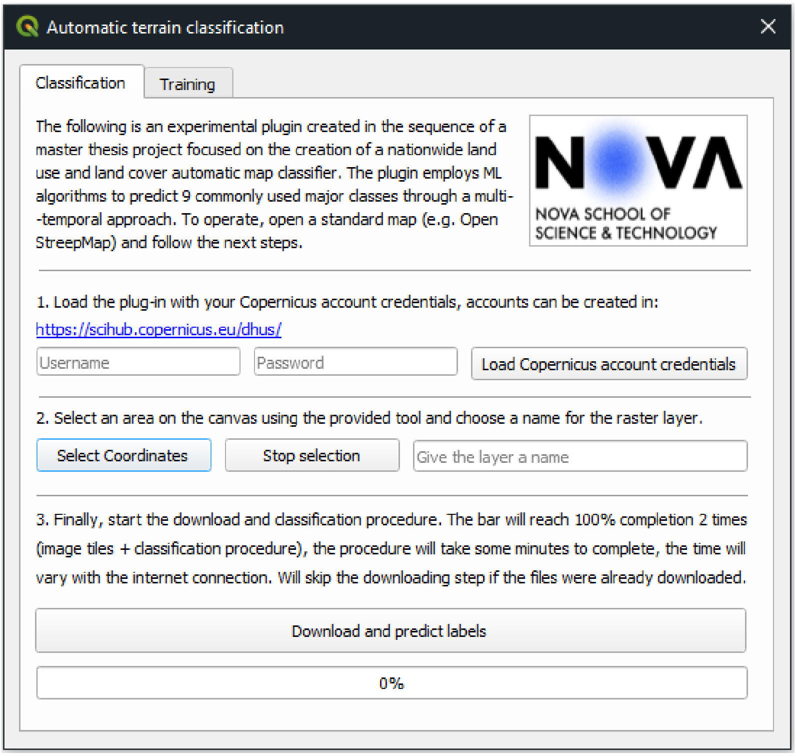
[158] D. P. Kingma and J. L. Ba. "Adam: A method for stochastic optimization." In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2015. arXiv: 1412.6980. URL: https://arxiv.org/abs/1412.6980v9.

Appendix A

Appendix 1 - Plug-in showcase

Throughout the body of the document, the creation of a QGIS plug-in was regularly mentioned. The aforementioned software, in line with the document title, takes the name of "Automatic Terrain Classification". The creation of the plug-in was paramount in the study of the proposed solution, allowing for gathering of all the employed satellite data and the creation of the data sets. Most of the code used in order to accomplish the various steps described in the document is still present, although inactive, in the form of functions in the plug-in files, as represented in figure A.2b. The code was preserved in order to provide a solid framework from which future attempts and novel methods may be applied in a repeatable environment, allowing for the accurate comparison between novel and previous approaches. In addition, as one of the major focuses of the solution, the plug-in offers the ability to employ the pursued approach to automatically classify any terrain area surveyed by the sentinel-2 mission, as seen in figure A.2a. Further information regarding the software setup guide, instructions and download links, can be found in the following GitHub repository URL: `https://github.com/jmsassuncao/auto_terrain_classification`

Figure A.1: *Automatic Terrain Classification* plug-in logo.


Automatic terrain classification
Classification
Training
NOVA
NOVA SCHOOL OF SCIENCE & TECHNOLOGY
Username
Password
Load Copernicus account credentials
Select Coordinates
Stop selection
Give the layer a name
Download and predict labels
0%

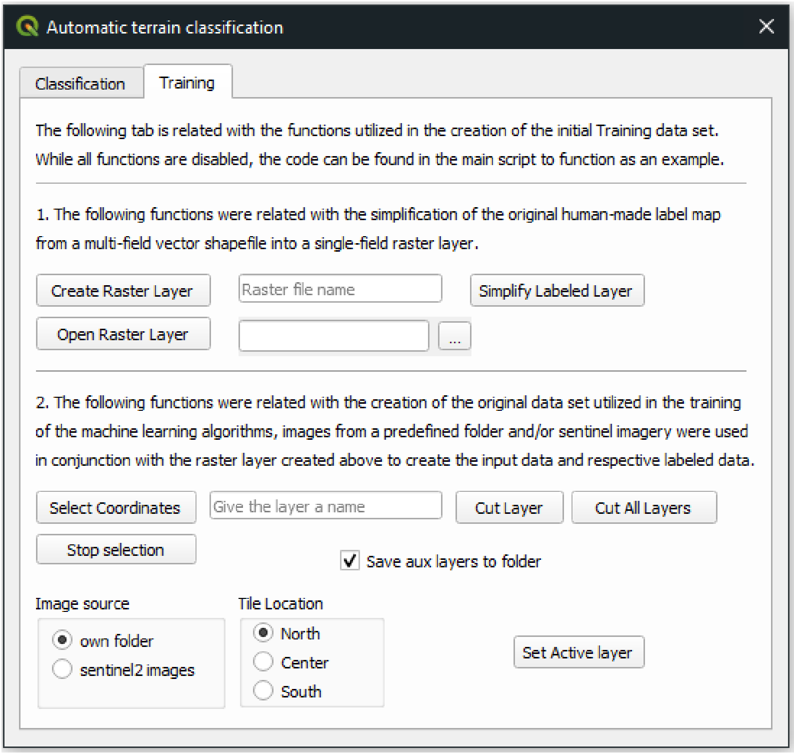
The following is an experimental plugin created in the sequence of a master thesis project focused on the creation of a nationwide land use and land cover automatic map classifier. The plugin employs ML algorithms to predict 9 commonly used major classes through a multi--temporal approach. To operate, open a standard map (e.g. Open StreepMap) and follow the next steps.

1. Load the plug-in with your Copernicus account credentials, accounts can be created in: https://scihub.copernicus.eu/dhus/

2. Select an area on the canvas using the provided tool and choose a name for the raster layer.

3. Finally, start the download and classification procedure. The bar will reach 100% completion 2 times (image tiles + classification procedure), the procedure will take some minutes to complete, the time will vary with the internet connection. Will skip the downloading step if the files were already downloaded.

(a) Plug-in classification tab.



The following tab is related with the functions utilized in the creation of the initial Training data set. While all functions are disabled, the code can be found in the main script to function as an example.

1. The following functions were related with the simplification of the original human-made label map from a multi-field vector shapefile into a single-field raster layer.

2. The following functions were related with the creation of the original data set utilized in the training of the machine learning algorithms, images from a predefined folder and/or sentinel imagery were used in conjunction with the raster layer created above to create the input data and respective labeled data.

☑ Save aux layers to folder

Image source
● own folder
○ sentinel2 images

Tile Location
● North
○ Center
○ South

(b) Plug-in data set creation tab.

Figure A.2: *Automatic Terrain Classification* plug-in GUI.