Universiteit Leiden

A Deep Learning & Computer Vision Based Approach to Airborne Laser Scanning data

Automated Instance Segmentation of Celtic Fields in LIDAR data from the Veluwe, Netherlands, using Mask R-CNN

Sohini Mallick (s2412071)

Course: Master Thesis Archaeological Science
Course Code: 1084VTS
Supervisor:
Dr. Karsten Lambers
Faculty of Archaeology
Leiden University

Final Draft

Leiden, June 2021

Preface

Having written this thesis in its entirety during COVID-19, I can say with conviction that this thesis could not have come to fruition without the support, motivation and guidance of numerous people. I owe this milestone to all of them. I would like to thank my supervisor, Dr. Karsten Lambers, for his patience and diligent feedback in the duration of this work. His insights were paramount in propelling this thesis forward at all times. I have the utmost gratitude towards Wouter Verschoof-van der Vaart, for allowing me to work on data from his project, giving me his time while working on his own PhD work and helping me out of more than one jam!

Like all endeavours in my life, I owe a 100% of my success in finishing my Masters degree and this thesis to my family. Despite being in a different continent, their overwhelming encouragement and absolute belief in me was the pillar of strength that held me up throughout this time. I would also like to thank my friend Abhay Murjani for being my go-to person for everything data science and endlessly and patiently giving his time answering my questions on anything and everything.

Lastly, I am extremely grateful for my friends in the Netherlands. Being so far from home in the middle of a pandemic took its mental toll, but our times together was more often than not the catalyst that pushed me to carry on.

Contents

Chapter 1

Introduction

1.1 Overview

For some decades now, archaeology has been subjected to a *data deluge* (Bevan 2015). This refers to an enormous influx of archaeological information in the digital format. One major source for this is remotely sensed spatial data, which has undergone numerous technological developments such as improved resolutions of commercial satellites and an increase in geographic coverage of image-based remote sensing data (Bevan 2015, 1474-1475). Analysing all of this image data manually has become an increasingly daunting task, particularly when considering the increasing complexity of the imagery. It is thus no surprise that much research in recent times, focuses on the development of new methodologies, particularly digital, for the analysis of image based remote sensing, in order to make the process more cost and time efficient (Opitz and Herrmann 2018; Lambers 2018).

Since the 1970s, computational tools have been developed for remote sensing imagery analysis, within the disciplines of geodesy, cartography and earth observation (Lambers 2018, 114). These approaches however differ from archaeology, as they concentrate on entire landscapes as opposed to selective traces of human activity of archaeological importance (Lambers 2018, 115). Analysis of remote sensing from an archaeological perspective can be done by automating the detection of specific archaeological objects and features in such data. Doing so allows a more efficient allotment of human expertise and resources to more domain specific interpretive tasks, such as connecting

the presence of these archaeological traces to wider human-landscape dynamics of the past. Research on the same has been conducted since the early 2000s, through interdisciplinary approaches of computer and earth science (Lambers 2018, 115).

One such approach is the use of Artificial Intelligence. Concepts from the discipline have been formerly used for archaeological research, with applications ranging from identification and reconstruction of ceramic sherds, determination of sex and anatomical properties from human remains, management of exhibitions in museums to identification of archaeological sites and features from various landscapes (Mantovan and Nanni 2020). The primary idea driving this thesis project is to use artificial intelligence based computer algorithms to create a methodology for the automated detection of archaeological objects from airborne remote sensing data, test it against an archaeological case study and evaluate its role from the wider perspective of archaeological prospection and landscape archaeology.

1.2 Methodological Context

1.2.1 Airborne Remote Sensing in Archaeology

"*Remote sensing* is the science and art of obtaining information about an object, area, or phenomenon through the analysis of data acquired by a device that is not in contact with the object, area, or phenomenon under investigation." (Lillesand and Kiefer 2015, 1). As per European terminology, the definition of this includes the use of the airborne and satellite technology, but not near-surface non-invasive geophysical prospection. Remote sensing techniques of the spaceborne and airborne variety have been the most popular form within archaeology, since around 1900 (Luo *et al.* 2019, 2). Its non invasive nature and ability to provide a wider look at the landscape from above has rendered it especially useful for analysis and detection of objects and features of archaeological importance. There has been a growing prominence of airborne remote sensing in archaeology in the past few decades, and as a result almost an explosion of large amounts of data of varying qualities, new technologies for data collection, sensors and so on (Opitz and Herrmann 2018; Lambers 2018; Agapiou *et al.* 2014; Chen

et al. 2018). Prominent technical developments been discussed below. It should be noted that this non-exhaustive overview focuses on only different imaging technologies, as the role of LiDAR as a data source is significant with respect to the archaeological case study (discussed in subsequent sections). Airborne remote sensing technologies also differ in terms of platforms (satellite-based, low altitude, Unmanned Aerial Vehicles), sensors (active and passive) and image format (digital format) (Lambers 2018).

The first use of remote sensing for archaeological purposes was brought about by *aerial photographs* acquired for the purposes of military reconnaissance and mapping, during the First and Second World War (Leisz 2013, 12). Crawford (1923) used aerial image data from Southern England to identify crop marks on cultivated land, otherwise undetectable from the ground. This was the first archaeological study that highlighted the potential in using aerial survey imagery for archaeological mapping. Aerial photography however, often fails to detect important archaeological landscape proxies such as crop marks and soil marks (Luo *et al.* 2019, 6). This is because detection of these features in photographs often requires specific weather, landscape and phenological conditions, thus making the technology limited in its efficacy (Luo *et al.* 2019). The advent of *hyperspectral/multispectral* imaging technologies, in combination with dimensionality reduction techniques such as Principal Component Analysis (PCA), helped mitigate this issue to some extent (Luo *et al.* 2019; Aqdus *et al.* 2012). Their ability to detect bandwidths outside the visible spectrum allowed enhancement of soil moisture stress in growing plants, which is linked to the appearance of crop marks (Aqdus *et al.* 2012).

The aforementioned are 'passive' technologies i.e. they simply capture reflections and radiations of the Earth's surface. In contrast, active technologies such as *Synthetic Aperture Radar (SAR)* and *Light Detection and Ranging (LiDAR)* actively send signals to the Earth's surface and then re-capture the partially reflected wave (Lambers 2018, 113). Thus, they can overcome limitations relating to weather conditions and nighttime acquisition, ability to penetrate through forest cover and greater accuracy in sensing buried features, soil-marks and archaeological micro-relief, by creating Digital Elevation Models,

as compared to passive sources (Luo *et al.* 2019, 11-14).

Remote Sensing provides a significant advantage over traditional methods of archaeological prospection due to the sheer amount of area that can be surveyed and analysed in a small amount of time. It also provides a much wider perspective at the landscape. This has proven particularly useful when studying complex networks, such as road and quarry systems or determining the extent of historical landscapes (Traviglia and Cottica 2011; De Laet *et al.* 2015). In fact, previous research has shown an ability of remote sensing techniques to even penetrate through submerged environments (Traviglia and Cottica 2011). Moreover, different categories of information from remote sensing data can have different archaeological implications. For example, a remote sensing project investigating roads and quarry systems in the Middle Egypt region showed that while the data generated provides an efficient way to visualise spatial context, the spectral information also gave information on road characteristics (Traviglia and Cottica 2011). Undoubtedly though, there are also certain challenges which require consideration and reflection, such as access to relevant and useful data, management of databases and archival responsibilities, and improving integration of datasets acquired through different sources (Opitz and Herrmann 2018).

1.2.2 Dataset Type: LiDAR

The term LiDAR is an acronym for Light Detection and Ranging. It is an active remote sensing system which transmits a pulse of energy to the Earth's surface, records the amount of time it takes for the signal to return through a sensor, and determines ranges or distances. The technology can be connected to either ground systems such as Terrestrial Laser Scanning, or airborne ones, such as Airborne Laser Scanning (ALS). The latter refers to LiDAR systems which can be mounted on aerial vehicles such as aircrafts and helicopters and is currently used extensively for archaeological application. In more recent research, drones equipped with LiDAR technology have also been employed (Risbøl and Gustavsen 2018).

The application of airborne LIDAR remote sensing techniques has

surpassed more traditional mapping approaches due to the digital elevation models that can be created from LiDAR collected data. Through the highly detailed raster imagery derived from 3D point clouds, entire landscapes can be viewed in 2D, 2.5D or 3D format, giving an extensive look at the topography, elevations, vegetation and built structures. It has proven to be more cost effective and displays significantly more amount of information, particularly in hilly and forested terrains with dense canopies, in a very short amount of time (Chase *et al.* 2012; Evans *et al.* 2013).

In certain areas of research, such as Mesoamerican archaeology, the use of LIDAR over other remote sensing techniques caused a noticeable difference. Past remote sensing based research was able to cover only smaller sample sites, instead of being able to conceptualise settlements in terms of the overall landscape (Chase *et al.* 2012). This is because other techniques cannot effectively filter out dense vegetation cover and forest canopies. This property is also what makes LiDAR an effective remote sensing tool for the Veluwe region, which is laregly under forest cover. Figure 1.1 and Figure 1.2 show a 2D and 2.5D (respectively) LIDAR Digital Terrain Model (DTM) of a research area from the archaeological site of Caracol, Belize. Through these, we can see the success of LIDAR in detecting not only the more easily visible and larger architecture, but also smaller features such as storage units, traces of agricultural terracing and looted burial chambers, after filtering out larger surface features (Chase *et al.* 2012, 12917-12919).

Past projects have shown how LIDAR can contribute to the analysis of micro-topographic earthworks and ancient landscapes. DTMs can allow the detection and analysis of different paleoenvironmental features. We see an example of this in the case of a natural park at Boscodell'Incoronata (Southern Italy) (Coluzzi *et al.* 2010).

One of the main factors of archaeological investigation is that in studies relating to the landscape, it is not enough to see it as a static entity. Rather, it is equally important to assess the dynamic systems associated with it, such as growth and decline of ancient urban centres and human-environment interactions. Research has shown that LIDAR collected data shows potential in providing better under-

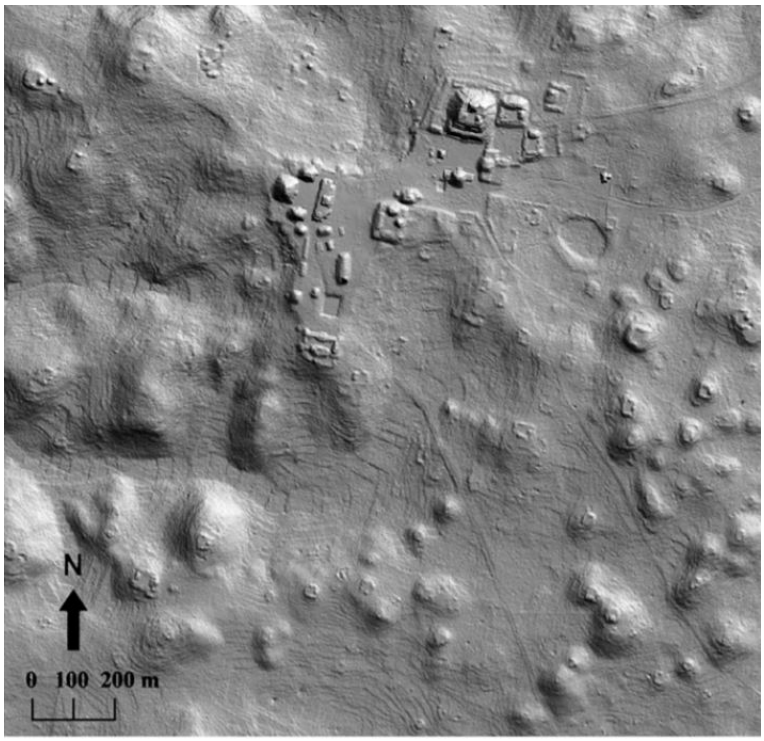


Figure 1.1: 2D LIDAR DEM of central Caracol(Chase *et al.* 2012, 12918)

standing of socio-economic models. Due to an absence of well defined spatial parameters (based on archaeological evidence), it was difficult to connect information from iconography or epigraphy, to socio-political models in Mesoamerican archaeology (Chase *et al.* 2012). LiDAR DTMs provided the necessary data required to make these comparisons. In the Khmer empire, road networks and hydraulic systems were unparalleled in the pre-industrial world, and caused repeated transformation of landscape for over a millenia, at a regional scale. These networks were identified in LiDAR imagery (Evans 2016).

1.2.3 Deep Learning approach to analysis of LIDAR data

The main methodology of this project is based on two concepts - *Deep Learning* and *Computer Vision*. Deep Learning is a type of Artificial Intelligence algorithm which uses 'neural networks' to predict a desired result on the basis of relevant input data, without human interference. A neural network in turn is a computational learning system which has been inspired by the way the *human brain* functions, through the interaction of connected neurons. A neuron takes

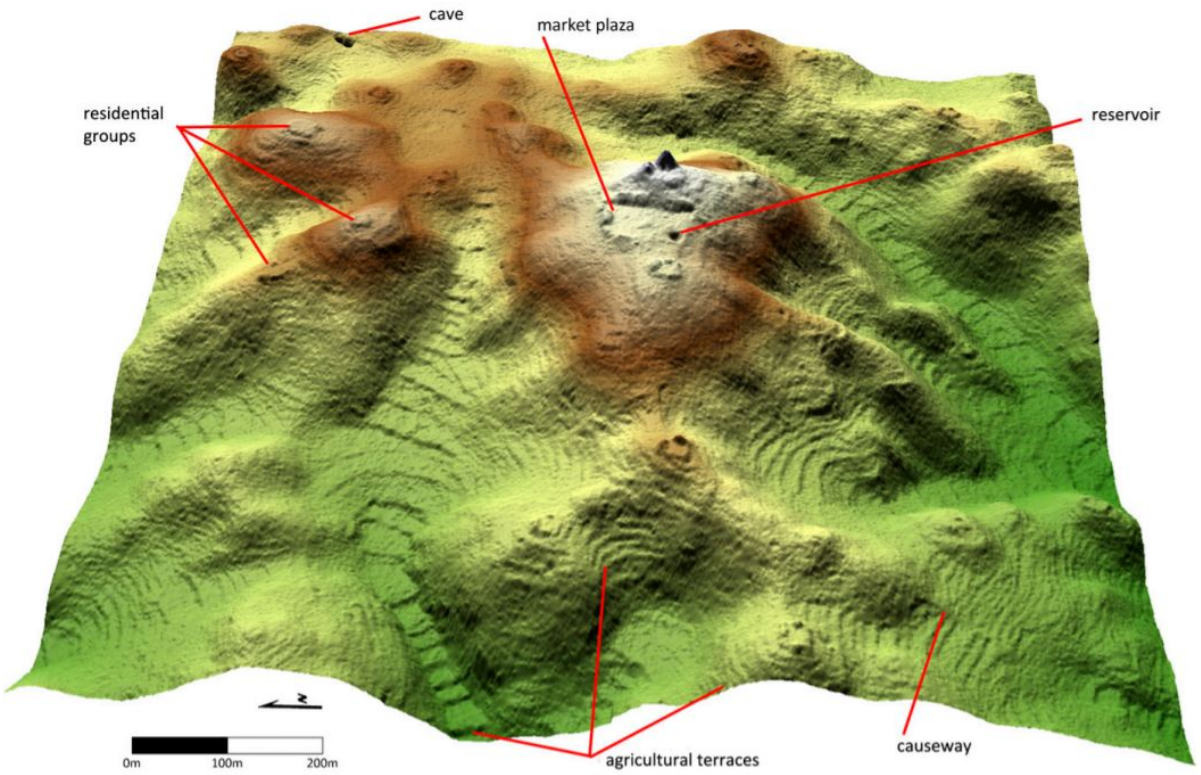


Figure 1.2: 2.5D LIDAR DEM of central Caracol(Chase *et al.* 2012, 12919)

multiple inputs, performs a function on them and sends the final result to an output.

Computer Vision, in accordance with its name, is an attempt by the computer science discipline to replicate the *human vision* and train computers to process and analyse data from images and videos. (These concepts will be explained in more detail in the 'Research Background' chapter).

The data deluge in archaeology, specifically with respect to remote sensing opens up various possibilities with respect to archaeological research. The uncertainty in archaeological data, combined with its density, highlights the need for computational methods in its treatment. Manually analysing such large amounts of data is a cost and time inefficient process. Aside from the quantity of the data, one must also take into account the different imaging technologies developed for remote sensing data collection, leading to a vast difference in quality, resolution and spectral information. A machine can computationally

update itself to adhere to the changes in quality and type of data, thus possibly providing a more efficient way of recognising relevant information data, in comparison to human visual judgment based on previous experiences (Soroush *et al.* 2020, 2). However, other parts of the wider framework, post the identification of archaeological features, still require human expertise for analysis and interpretation, or a combination with other digital tools. This is also the case for this thesis project.

Automated detection techniques using Deep learning for LiDAR data is a fairly new research field, going back approximately 5 years, within the archaeology discipline (Verschoof-van der Vaart and Lambers 2019; Kazimi *et al.* 2019; Caspari and Crespo 2019; Verschoof-van der Vaart *et al.* 2020; Bundzel *et al.* 2020; Verschoof-van der Vaart and Landauer 2021; Guyot *et al.* 2021). These implementations centre around Convolutional Neural Networks or CNNs, a deep learning architecture used specifically to solve pattern recognition tasks and other computer vision based tasks in image data (O'Shea and Nash 2015). Its framework for classification and mapping tasks is useful for archaeological analysis. In particular, they have greater sensitivity in identifying more obscure patterns in image data than other comparative AI algorithms (Caspari and Crespo 2019, 1). This is due to their method of taking inputs in the form of matrices as opposed to single row vectors, allowing the algorithm to extract information from all adjacent pixels (Caspari and Crespo 2019, 1).

Past research has shown successful detections of objects and features within archaeological landscapes, using CNN based Deep Learning algorithms. A higher number of true positives (i.e. correct positive detection of objects) have been observed, and more notably for archaeological prospection, a lower number of false positives (i.e. an incorrect positive detection of objects). CNN based detections have also worked effectively for landscapes that may have undergone natural changes or disturbances such as ploughing or post-depositional processes (Soroush *et al.* 2020, 4). The general consensus is that while deep learning based approaches in archaeology are not a complete replacement for human expertise, they facilitate effective use of time and resources in the research of large and complex datasets (Soroush

et al. 2020; Verschoof-van der Vaart *et al.* 2020).

1.3 Archaeological Case Study

1.3.1 Research Area: The Veluwe

The research area consists of a largely forested area known as the Veluwe in the Gelderland province of Central Netherlands (Figure 1.3). The area comprises the largest Pleistocene push moraine in the Netherlands, extending to length and width dimensions of 50 km and 12 km respectively (Bakker and Meer 2003, 144). Large wooded areas in the region were cut down around the 8th to 10th century AD for the purposes of agriculture, obtaining open grasslands for farm animals and iron production (van der Heide *et al.* 2008, 206). The over-exploitation led to soil erosion and large-scale sand drifts. Part of the total forested region is also a national park, providing important economic benefits, specially in terms of tourism (Hein 2011). It is considered an important area ecologically as well, due to having a wide variety of plant and animal species as well its size (van der Heide *et al.* 2008, 206). The Veluwe currently holds one of the biggest clusters of archaeological objects in the Netherlands, which are currently well documented due to a number of archaeological projects carried out in recent times (Arnoldussen 2018; Bourgeois 2013). These include charcoal kilns, barrows, Celtic fields, and hollow roads (Verschoof-van der Vaart and Lambers 2019; van der Heide *et al.* 2008).

The area is characterised by the covering of *plaggen soil* over the landscape. From the Medieval Period to the 19th century, the agricultural practice of stripping heather or grass sods and using them as bedding material in stables was done within the Pleistocene areas of northwestern Europe (Groenman-van Waateringe 1992, 87). This led to an artificially raised layer of soil, with a 50 cm thick humic topsoil, consisting of sand, clay and loam (Groenman-van Waateringe 1992, 87). Modern agriculture, involving the digging of this plaggen topsoil to reach the coversand underneath destroyed large amounts of the archaeological features (Groenman-van Waateringe 1992, 87). As a result, much of the archaeological traces in the region have low visibility and thus, are less identifiable using traditional survey methods.

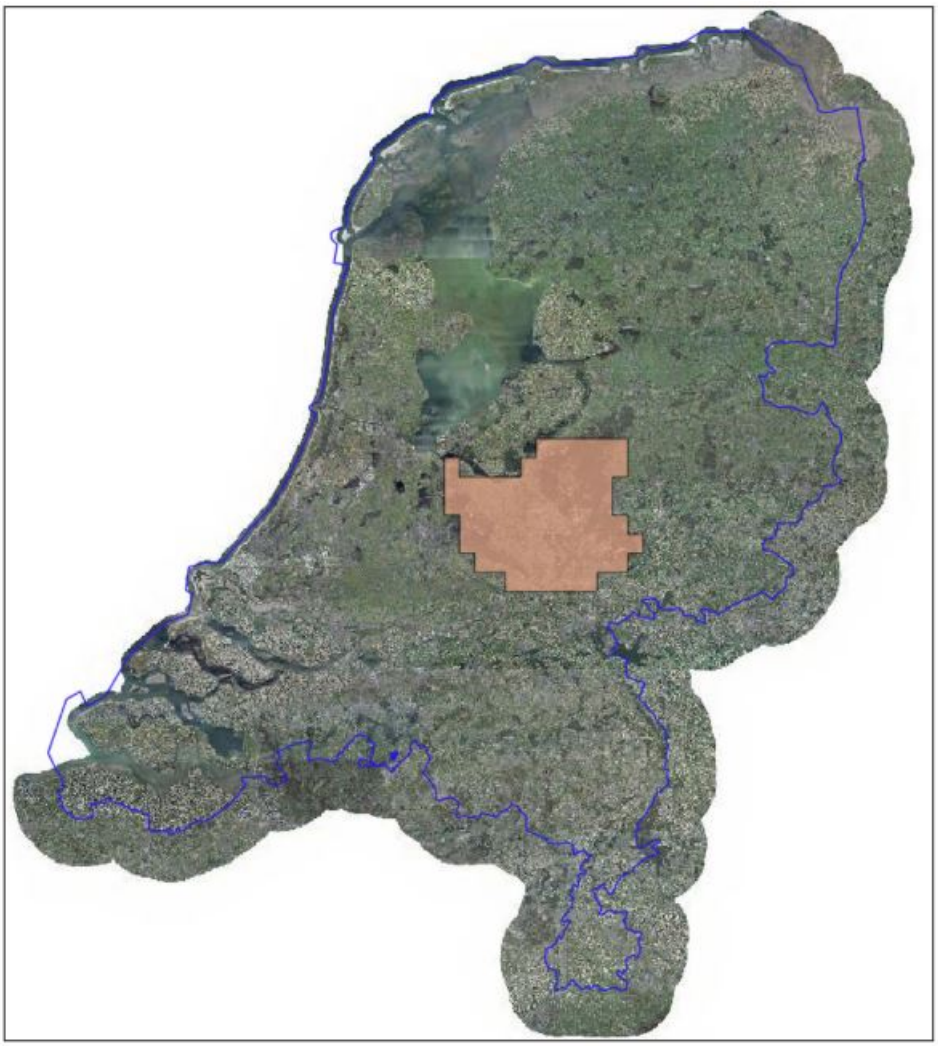


Figure 1.3: Research Area, highlighted in red (Verschoof-van der Vaart and Lambers 2019, 32)

Computational models have been used in the past to study cultural and environmental dynamics in the Veluwe. An example is the modelling of climate change for the purposes of forest management, using the Bayesian update theory (Yousefpour *et al.* 2015). Historical routes and path-networks have also been studied using a multi modelling approach (Vletter and Van Lanen 2018). The research combined GIS based modelling techniques such as network friction and least cost path, with environmental, ecological, historical and ALS datasets to reconstruct the historical, complex route networks within the areas (Vletter and Van Lanen 2018).

1.3.2 Archaeological Object: Celtic Fields

Celtic fields are prehistoric field systems found widely within North-Western Europe, including Sweden, Denmark, the Netherlands, Germany and Ireland (Spek *et al.* 2003; Nielsen and Dalsgaard 2017; Whitefield 2017). They date approximately from the late Bronze Age up to the Roman Iron Age. The name 'Celtic' fields in itself is a misnomer, as these field systems have no connection to the named ethnic group, although being initially attributed to the Celts by British archaeologists (Curwen and Curwen 1923).

Within the Netherlands, research on Celtic fields or *rattakers* involved a number of initial theories as to their function. These ranged from categorisations as military encampments, sheep-management areas and funerary ritual sites, before finally being identified as prehistoric field systems by Albert E. Van Giffen (Giffen 1939 in Arnoldussen 2018, 3). Most of them are located in the northern and southern parts, with the province of Drenthe in the north being a focus of past studies. With the advent of the digital elevation model database - the *Actual Height Model of the Netherlands (AHN)*, combined with ALS technology, more Celtic field systems were identified in the central part of the Netherlands (Kooistra and Maas 2008, 2320). This area had been previously neglected, due to the large amounts of vegetation and woodlands. Due to the use of LiDAR derived AHN data, almost 1500 ha more of surface area covered by Celtic fields was discovered, as compared to before (Kooistra and Maas 2008, 2327). In geomorphological terms, they were mainly found in the slopes of ice-pushed ridges, formed during the Saalian glacial period (Kooistra and Maas 2008, 2323).

Celtic fields consist of a number of small, square or rectangular adjacent plots, which form a larger rectangular checkerboard pattern on the landscape. Figure 2.4 shows an artist's impression of these Celtic fields (as presented by Arnoldussen (2018)). The individual plots are approximately 20 to 40 m wide and are separated from each other by sandy ridges or stone walls (Spek *et al.* 2003, 142-143). Due to disturbances caused by natural and man-made processes, traces of these fields are often not visible, even from a close distance. The use of

LiDAR on landscapes that contain Celtic Fields has allowed a better look at the extent of wider and bigger field systems. Research so far on them has included prospection (through aerial photographs), mapping (through various geodetic methods), excavations, study of landscape and land-use history through paleobotanic and stratigraphic studies (Spek *et al.* 2003, 143). However, they are still vastly understudied due to their general lack of visiblity.

Figure 1.4: Celtic Fields (Arnoldussen 2018)

The research undertaken by Arnoldussen (2018) extensively reviews the importance of studying Celtic Fields for archaeology in the Netherlands. These are the most time sustainable historical field systems in the Netherlands, preserved for over 400-600 years, through a mutltitude of changing landscape phases and agricultural practices (Arnoldussen 2018, 19). They show morphometric similarity across regions in the Netherlands, which leads to a wider understanding of field systems in the area (during this time period).

As an archaeological feature, they are relatively underused in the LIDAR automation research. The only implementations found (so far) while researching for this thesis is a Kriging based filtering technique applied on a case study from Doorwerth (Netherlands) (Humme *et al.* 2006) and one use of CNN-based object methodology by Verschoof-van der Vaart and Lambers (2019).

1.4 Research Goals

The use of Deep Learning for automated detection has shown largely successful results within archaeological research. However, these have concentrated on more uniformly shaped, localised and compact objects, such as barrows, charcoal kilns, mounds etc, which are easier to detect due to their morphology. Detection of more complex, interconnected landscape systems such as road networks and field systems are rare, as discussed by Traviglia and Torsello (2017). As dicussed by Davis (2021) discusses, it is important that a shift be made to detection of more widespread and complex patterns of human activity and landscape transformation, in order to fully utilise automated remote sensing for answering wider archaeological questions relating to cultural activity. A notable example is the reconstruction of hollow roads in the Veluwe from LiDAR data by Verschoof-van der Vaart and Landauer (2021). This methodology also used a Deep Learning based approach to detect and highlight an interconnected system of road networks, which could contribute to the understanding of transport networks in the Veluwe's post-Medieval period.

Celtic Fields are one such example of broader archaeological systems, which are relatively understudied in the archaeological record. Celtic Fields are one of the major large scale archaeological systems in the Veluwe region. Reconstructing these large-scale field systems could allow us to understand important agricultural practices in the region. Due to these reasons, they were selected as the archaeological case study for this project. By embedding an automated detection methodology of Celtic Fields into a wider framework consisting of other sources, we can gain important information about the role they played in the archaeological landscape of the Veluwe.

The data for this thesis was collected for a PhD project by Wouter Verschoof-van der Vaart, conducted as part of the Data Science Research Programme at the Faculty of Archaeology, Leiden University. This initiative is a collaborative effort to combine different fields of the humanities and sciences with data science technologies. The project involves using Deep Learning technologies for the automated object detection of archaeological objects from LiDAR data within the Veluwe region in the Netherlands (Verschoof-van der Vaart and Lambers 2019; Lambers *et al.* 2019; Verschoof-van der Vaart *et al.* 2020; Verschoof-van der Vaart and Landauer 2021). This also includes the detection of Celtic Fields. However, a different methodology is applied in this thesis project, and one of the motivations is to compare the performance of the two methodologies for this task.

Deep Learning comprises of different types of techniques for different tasks. The one used for this project is *instance segmentation*, using a state-of-the-art method called *Mask R-CNN*. An instance segmentation approach gives the exact outline of an object within a detected area of interest (Kazimi *et al.* 2019). This would be an advantage in the case of detecting and mapping out objects in the landscape which are non-discrete and non-uniform in shape. Therefore, it is hypothesised that this methodology will prove fruitful in the mapping of individual plots within the Celtic Field systems. This is especially due to the fact that the boundaries between plots often appear blurred on the input data, due to landscape changes such as erosion or human activity. The methodology will comprise of literature review and a practical application of the Mask R-CNN technique on the research data.

The main methodological research aim of this thesis is to evaluate the use of this instance segmentation technique for Celtic Fields. However, it will also be assessed in terms of how the results of this method can contribute to archaeological prospection and wider landscape archaeology research of the region.

The main research question that will be explored is: *How does a Deep Learning based Mask R-CNN algorithm perform with respect to the instance segmentation of pre-historic Celtic Fields, from remotely sensed LIDAR data of the Netherlands and subsequently con-*

tribute to archaeological prospection?. To answer this, the following sub-questions have been defined:
1. To what extent can the model identify instances of individual Celtic Field plots and delineate between them?
2. In what way do the results compare to a comparative object detection approach, previously applied on the data of the region?
3. How can the results obtained from the methodology contribute to archaeological interpretations of Celtic Field systems in the Veluwe region?

1.5 Thesis Structure

In the introductory chapter, a basic research background was given, outlining the process that led to the development of this research, and aims and research questions were discussed.

Chapter 2 titled 'Research Background' expands on the overview given in the introduction. It consists of a brief non-exhaustive review of past research on automated detection in archaeology, explanation of technical aspects of the methodology, results of the comparative methodology previously used on the dataset and finally some critiques present in archaeological literature with regards to the role of Deep Learning based techniques for automated detection.

Chapter 3 outlines the complete methodology used for this project. This includes a comprehensive review of the data. It also covers the steps undertaken to eliminate some limitations of the methodology for archaeological datasets, the practical experiments conducted and the metrics used for evaluation. Finally, the chapter ends with a section on integration of the method into a wider workflow in the future, that could help in contextualising the results better.

Chapter 4 presents the results of the experiments conducted and explanations for the same.

Chaper 5 comprises a discussion of the results obtained in the previous chapter. It also covers the significance of these results in terms of methodological evaluation and comparative research. In addition, the results are assessed in terms of archaeological research and the role played by this methodology for assisting archaeological prospection.

Finally, some of the limitations of the technique are also highlighted.

Chapter 6 is the last chapter of this thesis which contains conclusions formed on the basis of research results and answers to the research questions defined at the start of the thesis. In the end, future scope of the study is also highlighted.

Chapter 2

Research Background

In the first chapter, we touched upon the hypotheses and reasoning that drove this research. This section expands on the methodological and theoretical research background. Technical theory used has been reviewed here, with a more detailed description present in Appendix B. The section also covers a literature review of comparative methods used in the past.

2.1 Automation in archaeological remote sensing

The idea of building automated systems for detecting archaeological features in remote sensing and satellite data, using digital tools, dates back to the 1990s. This paradigm followed a success in the use of computational methods for image enhancement and processing in aerial photography & remote sensing (Redfern 1998a). Examples of earlier foray into this sub-field can be seen in the research done by Lemmens for the detection of circular objects (Lemmens *et al.* 1993) and Redfern's development of a morphology & topography based automated classification system for archaeological monuments (Redfern 1998b), using various mathematical and computational concepts. For an overview of papers relating to the use of automated detection techniques in archaeological remote sensing (up to 2016), refer to Lambers and Traviglia 2016.

2.1.1 Comparative Techniques

In the past decade, this topic of study has gained much prominence, with new techniques and methodologies being proposed constantly. It is out of the scope of this thesis to discuss all of these new developments, however two significantly categories have been reviewed. An overview of research papers on various automated detection methods can be seen in Lambers *et al.* 2019.

1. Geographic Object-Based Image Analysis: The advent of GEOBIA or Geographic Object-based Image Analysis brought about a shift in automated detection techniques from predominantly pixel-level computation, to a contextual object oriented approach. A formal definition of GEOBIA, as given by Hay and Castilla (2008) is "...a sub-discipline of GIS devoted to developing automated methods to partition remote sensing imagery into meaningful image-objects, and assessing their characteristics through spatial, spectral and temporal scales, so as to generate new geographic information in GIS-ready format." (Hay and Castilla 2008, 77).

   GEOBIA follows a segmentation based-approach i.e. it consists of mathematically partitioning the image in order to find significant pixel clusters (as opposed to computing information from each individual pixel). Subsequently, a rule-set is defined to perform classification within these meaningful segments. The most common software used for GEOBIA processing is the eCognition software (Trimble Germany 2011).

   Around the 2010s, the implementation of GEOBIA on LIDAR datasets increased. The method when tested with multiple approaches to fossil landscapes in Italy and detection of barrows in Spain showed great promise in the mapping of ground level features and detection of sites in previously unsurveyed areas(De Guio *et al.* 2015; Cerrillo-Cuenca 2017). However, limitations were seen in accounting for depositional & post-depositional ladscape process, detecting objects in the 'border' regions of the input images & a high number of false positives (Trier *et al.* 2009; De Guio *et al.* 2015; Cerrillo-Cuenca 2017).

2. Template Matching: Template-based approaches involve manually isolating a representative section of the image, to form a 'template' of each object class. A correlation technique is then used over the dataset to search out comparable patterns (Cheng and Han 2016, 3. The limitations to this method are that it is a case-study sensitive method, thus having limited transference potential, and is also computationally expensive (Cheng and Han 2016). This is exemplified in Boer (2007)'s paper, wherein the author mentions that separate templates would have to be designed for detection of similarly shaped archaeological objects, such as burial mounds, village-mounds & coversand-dunes (Boer 2007, 247). Applications have also revealed a high number of false positives as compared to true positives (Menze *et al.* 2006), however Schneider *et al.* (2015) showed that these can be somewhat mitigated by other techniques, such as use of morphometric variables (Schneider *et al.* 2015).

2.2 Basic Theoretical Concepts

This section presents definitions of basic concepts imperative for understanding the methodology used in this thesis. As a whole, these concepts are sub-fields which belong to the wider discipline of Artificial Intelligence (AI). AI is a computer science discipline devoted training machines to perform tasks usually associated with human intelligence, without human intervention.

2.2.1 Deep Learning

Before diving into the intricacies of Deep Learning, we first need to understand the more commonly used concept of *Machine Learning*. Machine Learning is a branch of AI, that can be defined as 'training' computational algorithms to detect patterns in past data, and subsequently make decisions and predictions regarding new data (Alpaydin 2014), using mathematical models and statistics.

The process of creating and implementing a machine learning model can be roughly categorised into four steps - Data Collection & Preparation, Training, Evaluation/Inference and Tuning. The first step, i.e.

Data collection and preparation is the collection of relevant data which is used to train the algorithm. It also includes the pre-processing step of "cleaning" the data and converting it into the required format. Ideally the dataset is divided into three sets - training, validation and test. A **training set** consists of the input data examples which the algorithm utilises for 'learning'. The **validation set** is an unseen part of the training set which is used to check the performance of the algorithm on the training data, and tune it accordingly. These are the two sample sets, used in the second stage i.e. *Training*. The *Evaluation/Inference* is done using the test set which comprises of new data, unused in the learning stage, to evaluate the predictive capabilities of the created model (Mohri *et al.* 2012, 4). Finally, the *Tuning* stage includes maximising model performance by optimizing the input parameters.

Machine Learning paradigms make use of a set of object proposals & feature representations, to perform feature extraction and dimensionality reduction on an input set, & output class labels for the objects within the image (Cheng and Han 2016, 10). They have shown an ability to play an important role in analysing archaeological research data, including numerical, textual, geospatial and image data (Bickler 2021). The use of the Random Forest algorithm for detection of Neolithic burial mounds showed that ML was able to overcome the specificity required for rule-based methods, therefore acting as a more robust system for a heterogenous archaeological landscape & also improved the ratio of true positives to false positives (Guyot *et al.* 2018). ML methods have also been successful in automated detection when applied on other datasets (Menze *et al.* 2006; Orengo *et al.* 2020).

Deep Learning is a subset of Machine Learning, which utilises deep architecture of Artificial Neural Networks (ANNs). These are computational systems, built to mimic the learning process of the human brain. It consists of a number of processors called neurons, which are interconnected to one another and collectively collect an input and optimise the output (Schmidhuber 2015, 86). A basic structure of ANN architecture can be seen in Figure 2.1. The significance of DL algorithms is that as opposed to general ML algorithms, they have the ability to extract relevant features from the data for training

by themselves, on the basis of architectural hyper-parameters, rather than requiring they be fed by the user. In that sense they are more of a 'self-learning' set of algorithms.

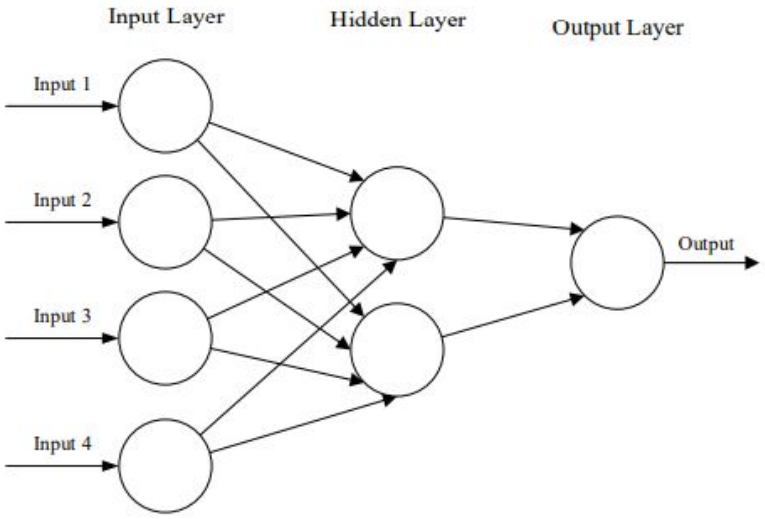


Figure 2.1: Basic structure of an Artificial Neural Network(ANN) (O'Shea and Nash 2015, 2)

2.2.2 Computer Vision

Computer Vision is a field of computer science which centers around the ability of computers to see and interpret digital images and videos. It computationally mimics all tasks performed by the biological vision system - including visually sensing stimuli, interpreting it and extracting relevant information to be used for other processes (Szeliski 2011).

Deep Learning applications of Computer Vision can be broadly categorised into three types - classification, detection and segmentation. Some examples can be seen in Figure 2.2.

1.*Image classification*: Consists of assigning class labels to objects within a complete image. The input image is labelled on the basis of probability that an object class exists within the image (Guo 2017, 38).

2.*Object detection*: Aside from indicating the presence of a specific class, object detection also localises the position of the objects within the image. This is done by generating a bounding box and class label around the object instance. The detection is considered a correct one

(a) Classification

(b) Object Detection

(c) Image Segmentation

Figure 2.2: Applications of Deep Learning in Computer Vision (Guo 2017)

if there is more than 50% overlap between the ground truth and detected boxes. This threshold can also be varied (Guo 2017, 41). The most commonly used dataset for object detection are the PASCAL VOC datasets (Everingham *et al.* 2010).

3. *Image Segmentation*: This process involves partitioning an image into multiple segments, so that each segment can be analysed more easily to detect an object. The difference between object detection and image segmentation is that in the latter, each pixel within the image is assigned a class label. This helps in identifying the image at the pixel level, and creating a binary mask which helps identify the exact shape of the object (Szeliski 2011).

2.3 Convolutional Neural Networks

To summarise the different comparative methods reviewed previously, rule-based algorithms see a large difference in the ratio of false positives to true positives, which is not viable for purposes of archaeological prospection. Moreover, these approaches have a more task-specific approach, which makes it harder to compare with other different objects or landscapes (Soroush *et al.* 2020, 2). They also find it difficult to account for landscape diversity and heterogeneous nature of archaeological structures within a landscape (Guyot *et al.* 2021). Machine learning approaches mitigate this to some extent. However, one of the biggest problems to overcome in computer vision is specifying the 'right' features for the machine to learn. Neural Networks have overcome this limit as self-learning algorithms, which extract relevant features from within the image itself, without relying on human assistance. This aspect specifically has a lot of potential for applications in archaeological landscapes, which are often vastly heterogeneous, in terms of objects and topography.

Convolutional Neural Networks or CNNs are the deep learning algorithms most commonly used for Computer Vision applications (for a more detailed review of CNNs and their architecture, see Appendix B). Their main tasks are pattern analysis and recognition within images, and they are built to handle the computational complexity of the same (O'Shea and Nash 2015). A CNN network is trained in two stages. The first stage, or the forward stage comprises representing the input image with the associated weights for each layer(Guo 2017, 10). A loss cost is then calculated by comparing the predicted output with the ground truth labels (Guo 2017, 10). In the second stage, or the backward stage, the loss costs are used to re-update the parameters, which is followed by a another stage of forward transmission. This process continues for a defined number of iterations (Guo 2017, 10). When the entire dataset completes one cycle of forward and backward transmission, it is known as an epoch.

A CNN comprises of three main layers - convolutional layers, pooling layers and fully connected layers. The first two layers perform the task of feature learning and are generally placed in alternating

layers. The last fully connected layers perform the task of assigning class labels to the input image (Voulodimos *et al.* 2018, 2). The basic structure of a CNN network can be seen in Figure B.3.

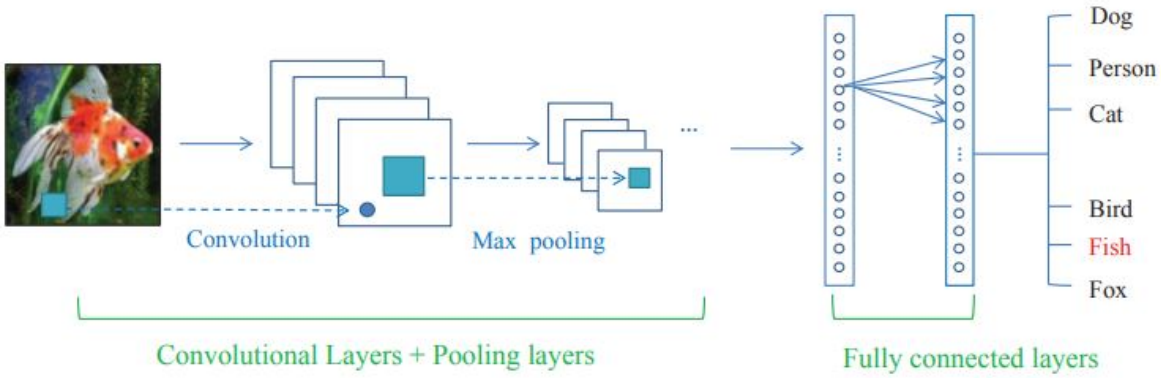


Figure 2.3: Basic structure of a CNN architecture

2.3.1 Object Detection

CNNs showed significant results with respect to image classification in 2012 (Krizhevsky *et al.* 2012), on the ImageNet Large Scale Visual Recognition Challenge (Deng *et al.* 2009; Russakovsky *et al.* 2015). However, object detection differs from classification as it also requires localisation of objects within an image. This an important requirement for archaeological prospection, as the position of a feature or object and its context within the landscape play an important part.

To solve this issue, R-CNN or Regions with CNN features was developed (Girshick *et al.* 2014). This method however is computationally expensive. Therefore Fast R-CNN was developed to reduce the computational time, and also improve detection. (Girshick 2015). Faster R-CNN further brings down computation time and makes the process relatively cost free (Ren *et al.* 2017).

In the last decade, a number of research projects have been implemented using CNN-based object detection approaches. These have been shown to successfully detect a number of archaeological objects on different types of landscapes, such as barrows, charcoal kilns, Celtic Fields & burial mounds (Verschoof-van der Vaart *et al.* 2020; Caspari and Crespo 2019. Verschoof-van der Vaart and Lambers (2019) &

Caspari and Crespo (2019) provide a comparison of their CNN based methodology to others, and show better performance metrics, as well as lower percentage of false positives (Verschoof-van der Vaart and Lambers 2019; Caspari and Crespo 2019). The former of these is significant in terms of the technical efficiency of the methodology, while the latter from the standpoint of archaeological prospection.

2.3.2 Automated object detection project for the Veluwe

The PhD research project at Leiden University on the automated detection of objects from Veluwe-focused LIDAR data consists of implementing an integrated methodology, including the use of remote sensing, CNN, citizen science and a domain-specific Location Based Ranking (at present), for archaeological prospection of the Dutch landscape. Initial research was on the use of Faster R-CNN to create a multi-class detector for barrows, Celtic Fields and charcoal kilns (Verschoof-van der Vaart and Lambers 2019). The first workflow developed, titled WODAN (Workflow for Object Detection of Archaeology in the Netherlands), was able to perform quite well in the detection of barrows and Celtic Fields, notably the former, in comparison to other detection methodologies. The predictions from the deep learning model were also assessed in a GIS environment during post processing to get a more 'real world' interpretation of the results (Verschoof-van der Vaart and Lambers 2019, 36). The results were subsequently validated through traditional fieldwork strategies and a citizen science initiative (Lambers *et al.* 2019).

A modified workflow (categorised as WODAN2.0) was designed to improve the model. This was done by first increasing the dataset, by adding more random data, including negative samples with no archaeological objects present in them. In addition, concepts of domain based predictive modelling was applied. This was done through a Location Based Ranking method, which made use of landscape characteristics, natural depositional process and human influence, and accounted for their impact on the visibility of archaeological objects (Verschoof-van der Vaart *et al.* 2020). The new workflow outperformed the previous one, showing the significance of using domain

knowledge (Verschoof-van der Vaart *et al.* 2020, 14). The results however did not still outperform those from the citizen science project.

Finally, a novel CNN algorithm was developed and combined with image processing techniques, to reconstruct hollow roads (Verschoof-van der Vaart and Landauer 2021). This was dubbed *CarcassonNet* and is a state-of-the-art methodology for the archaeological domain. The research followed an interesting approach, wherein the input data was fed as small sections of roads, rather than entire ones. This was done to account for the different geomorphological processes that had disturbed the landscape, and affected the orientation and morphology of the tracks.

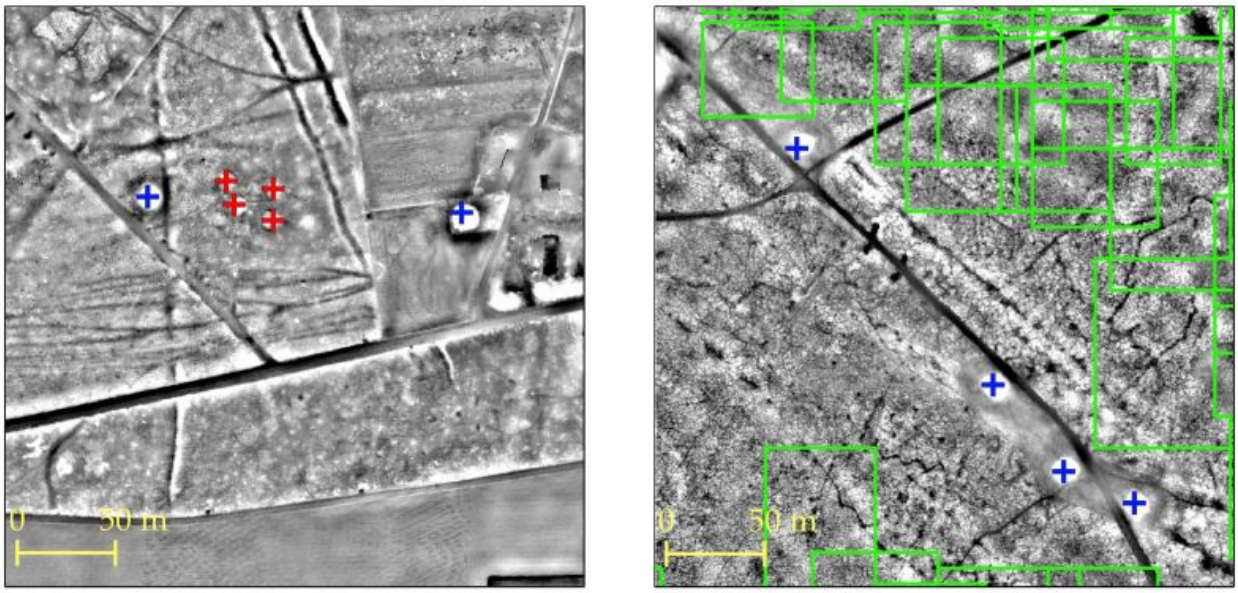


Figure 2.4: Results from the WODAN2.0 object detection workflow: Barrows - blue, Charcoal Kilns - red, Celtic Fields - green (Verschoof-van der Vaart *et al.* 2020, 15)

2.3.3 Image Segmentation

CNN-based object detection methods are limited in extracting accurate morphological information, by the strictly rectangular/square bounding boxes used. Post-processing techniques can be applied to mitigate the same, however a more efficient methodology would be to use image segmentation, which 'search' for objects in the image

through pixels rather than trying to identify the entire object. Although a pixel-oriented approach, it is not to be confused with pixel-based *image classification* methods that have been used in the past (Custer *et al.* 1986; Bennett *et al.* 2012; Lasaponara and Masini 2018). The latter consists of assigning a class to each pixel in image, in order to construct a thematic image. A major limitation of this method, with respect to archaeological prospection and landscape archaeology is that it does not account the contextual relationship between pixels and their environment. Therefore it is largely unfeasible for detection of archaeological objects. Image segmentation however takes extracts of relevant pixels within a partitioned image, thus making it more accountable for contextual information (for example the GEOBIA method is also an image segmentation method).

Image segmentation can be classified into two categories - semantic and instance. Past research using segmentation techniques have used both instance (Kazimi *et al.* 2019; Guyot *et al.* 2021; Bonhage *et al.* 2021) and semantic segmentation methods (Bundzel *et al.* 2020). A semantic segmentation algorithm clusters pixels within an image that belong together semantically, and embeds the spatial information of objects. Therefore, each pixel within the image is assigned a class label, rather than detecting complete objects. A common deep learning semantic segmentation architectures is UNet (Ronneberger *et al.* 2015). The method however does not take into account the general overall context in its pixel-wise, approach. Moreover, there is no instance-awareness of different objects of the same type (Garcia-Garcia *et al.* 2018, 9). In case of remote sensing imagery specifically, high levels of pixel accuracy is required, as almost every object contains meaningful information, thus causing problems in delineation of object boundaries (Yuan *et al.* 2021).

Instance segmentation is a method which tackles both object detection and semantic segmentation. It involves the prediction of object instances, AND producing their pixel-wise segmentation mask. It differs from semantic segmentation in that it delineates each individual object instance within a category (Figure 2.5). It is due to this property, that instance segmentation was chosen for Celtic Fields. In contrast to some common archaeological objects used in automated de-

tection methodologies, such as barrows, Celtic Field plots are present adjacent it each other as part of a wider field system in the landscape. An object detection method causes incoherence due to this structure, and a purely pixel-based approach such as semantic segmentation might also not be able to take into account properly the banked walls that separate the field. A combination of both allows properly separating the fields through detection, and providing more detailed information through the segmentation step.

(a) Input Image (b) Semantic Segmentation (c) Instance Segmentation

Figure 2.5: Types of Image Segmentation (https://analyticsindiamag.com)

2.3.4 Mask R-CNN

Mask R-CNN (or Mask Regional Convolutional Neural Network) is a state-of-the-art deep neural network developed for the purposes of solving instance segmentation problems. It was first proposed in 2018, as a framework which extends the Faster R-CNN object detection neural network, by adding an extra branch which predicts a binary mask for the object being detected, in addition to a bounding box and class labels (He *et al.* 2017). Thus the segmentation process occurs parallel to classification and detection. The framework consists of mainly three stages - extracting feature maps, generating ROIs through a *Region Proposal Network* and finally using the generated ROIs to perform instance segmentation and object detection, through a fully convolution network (FCN) (refer to Figure 2.6).

The following are the main elements that make up Mask R-CNN architecture:

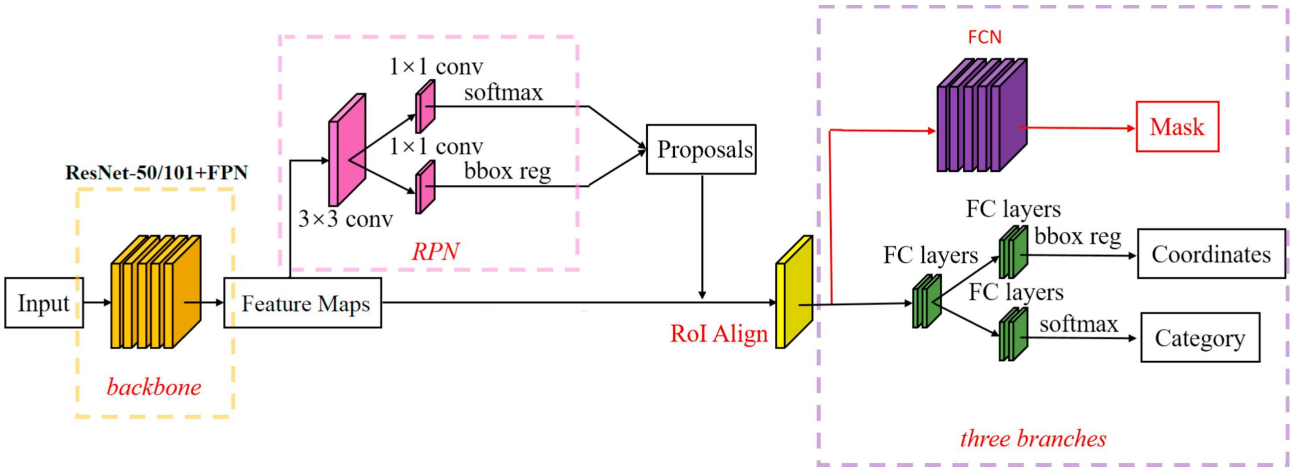


Figure 2.6: Mask R-CNN architecture, with ResNet+FPN backbone (Yu *et al.* 2019, 4

Backbone: ResNet+FPN

The 'backbone architecture' of a deep neural network refers to the starting convolutional layers used for initial feature extraction from the input image. Having depth in backbone network architecture is crucial for model performance (Simonyan and Zisserman 2014; Szegedy *et al.* 2015). However, it was discovered that adding more layers to the network also led to a degradation in model performance (He *et al.* 2015, 1). This problem was mitigated with the advent of *residual blocks*, used to create *Residual Networks* or *ResNet*. Mask R-CNN uses either *ResNet-50* or *ResNet-101* (50 and 101 layers respectively) backbone architecture (He *et al.* 2017).

In addition to ResNet, another backbone architecture called a *Feature Pyramid Network (FPN)* is also used. One of the main challenges in computer vision is accounting for the difference in scales in object instances (Adelson *et al.* 1983). Featurized image pyramids were conceptualised to solve this problem (Lin *et al.* 2017).

Stage I: Regional Proposal Network (RPN)

The RPN (Figure 2.7) stage of the architecture deals with the actual "detection" of object instances. It outputs a set of rectangular object proposals (regressor layer), along with a score evaluating object detection capability with reference to the background class, known as objectness score (classifier layer) (Ren *et al.* 2016, 3).

In Fast & Faster R-CNN, ROIPool operation is used for feature extraction (Girshick 2015; Ren *et al.* 2016). This method worked well for the object detection applications of these networks, however they were not aligned for pixel-to-pixel alignment. *ROIAlign* fixes this misalignment, by preserving exact spatial locations (He *et al.* 2017, 3). Replacing the ROIPool layer with ROIAlign showed up to a 50% increase in mask accuracy (He *et al.* 2017, 2).

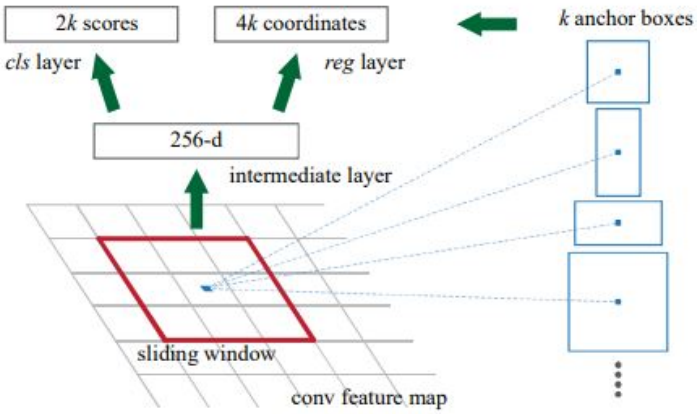


Figure 2.7: Regional Proposal Network (Ren *et al.* 2016)

Stage II: Network heads

The Box Head and the Mask Head are responsible for the actual formation of bounding boxes, the mask and predicted class labels (He *et al.* 2017, 3).

Guyot *et al.* 2021's implementation of Mask RCNN showed good results on a diverse landscape, with a number of heterogeneous archaeological structures, similar to the Veluwe. The research also shows how creating a mask outline outlining the shape predicted of predicted objects, can be used to go a step further from detecting the presence of these objects, to extracting further contextual and morphological information with respect to the actual landscape, such as area, perimeter and hyper-scale topographic signatures (Guyot *et al.* 2021, 7,11,16). The Mask RCNN method has also shown to provide good detection of sites in close spatial proximity, as well as in regions were the landscape has been disturbed by human activities (Bonhage *et al.* 2021, 7). In the project by Bonhage *et al.* (2021) specifically,

we can see high values of common metrics used for model evaluations, with very low number of false positives as compared to true positives (Bonhage *et al.* 2021, 6).

There were some limitations identified as well with respect to this methodology. One of these is the level of sensitivity observed to different combinations of images in the training dataset, for the same number of samples (Verschoof-van der Vaart *et al.* 2020; Guyot *et al.* 2021). In addition, all of the implementations of Mask RCNN have been successfully applied mostly to discrete and uniformly shaped objects, with clear landscape delineations, such as barrows, charcoal kilns and bomb craters (Kazimi *et al.* 2019; Guyot *et al.* 2021; Bonhage *et al.* 2021), as opposed to more complex, connected archaeological systems, with little spatial separation between instances.

2.4 Summary

This section started with a basic overview of the wider concepts of Deep Learning and Computer Vision. Thereafter, it was explained how Convolutional Neural Networks work, and how automated detection of archaeological objects in the past have benefitted from different CNN approaches.

The method chosen for this thesis is Mask R-CNN. Like most ML algorithms, using this technique consists of - collecting the data, feeding it into the architecture and training it for specific iterations. The output from the model on a test dataset will include a polygon mask covering an individual Celtic Field plot and mimicing its shape, as well as a class label. The next chapter covers the different steps that were required to effectively use the Mask RCNN model for the research dataset.

Chapter 3

Methodology

3.1 Requirements

This section covers the main hardware and software framework required for the implementation of this project.

NVIDIA GPU,CUDA & CUDNN: A Graphic Processing Unit (GPU) is a specialised electronic circuit designed to have a massive parallel architecture which can handle multiple tasks at the same time. To implement deep learning algorithms, the GPU runs the more expensive tasks, while the simpler ones are handled by the CPU. Using GPU implementation for deep learning computer applications has caused a significant increase in computational efficiency over CPUs (Shi *et al.* 2016).

CUDA is a parallel computing platform, developed by NVIDIA. It allows software development of codes that can utilise GPU systems, and contains debugging, profiling and compiling tools which could facilitate the same.

cuDNN (CUDA Deep Neural Network) is a GPU accelerated library, more specifically used for deep learning implementations. Using cuDNN alongside CUDA greatly accelerates and optimizes training and running of deep learning applications.

Python : All of the model training and inference was done using the Python programming language (www.python.org). The integrated development environment (IDE) used was PyCharm, of which a free and open source commercial version is available by the company JetBrains (www.jetbrains.com/pycharm).

Tensorflow : Tensorflow is an open source framework developed for deep learning applications in 2011, by Google (Abadi *et al.* 2016). It supports a number of programming languages, such as C++, Java, Javascript, Ruby and Swift, with Python being the most compatible. With Tensorflow, operations are performed on multidimensional arrays or 'tensors'. The Tensorflow package has many useful features, including fast computing speeds, flexibility to solve numerous complex mathematical problems, easy debugging through the Tensorboard tool, use of multiple GPUs for computation, proper documentation and support and is easy to use (Zaccone and Karim, 32). It is equipped with basic buildings blocks of deep learning systems such as different CNN layers and activation functions, as well as common off the shelf optimizers (Pang *et al.* 2020, 234-235).
Keras: Keras is a Python based high level API (Application Programming Interface), which works on top of a backend, such as Tensorflow or Theano. Keras enables fast experimentation, is user friendly and allows fast prototyping for CNNs. It can be used on both CPU and GPU systems (Nara *et al.* 2019). Keras can be used to access a number of pre-built functions important for constructing a CNN model. These include convolutional and pooling layers, dense layers (to construct a deep network), regularisation techniques to prevent overfitting and activation and loss functions. It can be used to improve the training process, by setting certain callback functions which stop training in between, based on certain metric considerations. It is also used to save the weights generated in the training iterations (Gulli 2017)

Table 3.1 shows the different versions used of these requirements in the implementation.

Table 3.1: Requirements with versions used for project implementation

GPU	GeForce GTX 1050 (4 GB)
CUDA	10.1
cuDNN	7.6
Python	3.8.5 (with Anaconda)
Tensorflow	2.3.0
Keras	2.4.3

3.2 Data

The input data manually fed into the CNN algorithm for training consisted of two elements: the LiDAR image data of the Veluwe region, as well as the 'annotations'. The latter consisted of a file, which outlined the location of the Celtic Fields in the images along with the label describing it as such. The CNN algorithm 'learned' the features of these Celtic Fields, by referring to the annotations, and then the subsequent model created used this knowledge to make predictions on new data.

This section covers the properties and metadata of the image data used, as well as outlining the process of annotating Celtic Fields within the same.

3.2.1 LiDAR Data

The LiDAR data was obtained from the open source online repositories PDOK (www.pdok.nl) and Actueel Hoogtebestand Nederland or AHN (ahn.arcgisonline.nl) by Wouter Verschoof-van der Vaart for his PhD research at the faculty, and subsequently also used for this project (Verschoof-van der Vaart and Lambers 2019; Verschoof-van der Vaart *et al.* 2020). The AHN is a digital elevation map, containing precise height data for the Netherlands. This file has been created using laser altimetry technology. Currently, two versions of the AHN exist: AHN1 was created between 1996-2003, with a point density of 1 point per 16 m. Subsequently in 2006, another study was carried out in order to update the existing data, and the AHN2 was created (van der Zon 2013, 6). Data from the Veluwe was carried out by the Dutch Directorate-General for Public Works and Water Management, and added to the AHN project named '2010-West' (van der Zon 2013, 27). Table 3.2 shows the properties and parameters of the LiDAR data used.

The AHN data downloaded consisted of LiDAR data with ground and non ground points, which was interpolated into a Digital Terrain Model. The interpolated data, in the form of Geotiff tiles, were visualised using the Simple Local Relief Model from the Relief Visualisation Toolbox (Zakšek *et al.* 2011; Kokalj and Somrak 2019) in a

Table 3.2: Properties of the LiDAR data used as define in (van der Zon 2013)

Meta-information LiDAR data	
Purpose	Water Management
Time of data acquisition	April 2010
Equipment	RIEGL LMS-Q680i Full-Waveform
Scan Angle (whole FOV)	45°
Flying Height above Ground	600 m
Speed of aircraft (TAS)	36 m/s
Laser Pulse Rate	100,000 Hz
Scan Rate	66 Hz
Strip Adjustment	yes
Filtering	yes
Interpolation Method	Moving planes
Point-Density (pt per sq m)	6–10
DTM-resolution	0.5 m
Vertical and Planimetric accuracy	0.05m

GIS platform. This was done in order to enhance local details of the research region by suppressing large scale terrain relief.

The tiles were then converted to JPG image files and sliced into smaller sub- images of size 600 by 600 pixels, resulting in a dataset containing 1024, 88 and 828 subtiles for the training, validation and test set respectively (Verschoof-van der Vaart *et al.* 2020). This was done since the initial size of the images was too big for a CNN algorithm to effectively process. Out of the total dataset, only images containing Celtic Fields were used for the project. Thus the final dataset used contained a training set of 147 sub-images, and 18 each for the validation and test sets. For future work, it would be interesting to add more 'background only' images i.e. images that do not contain the archaeological object of interest, and see if this has an improved effect on the ability of the model to distinguish between the background and object, and thus the mask delineation.

Table 3.3: Split between training set, test set and valisation set in the original dataset as well current project

Dataset	Total Images	Training Set	Validation Set	Test Set
Originally available	1924	1024	88	828
Mask R-CNN	184	147	18	18

3.2.2 Annotations

The dataset for the original PhD project consisted of annotations in the object detection PASCAL VOC XML format (Everingham *et al.* 2010). Initial experiments conducted with this format showed that it did not yield the results required from the instance segmentation methodology, in terms of outlining the shape of the Celtic Field plots with accuracy (Appendix D shows a prediction on the input data).


```
{"33an2_SLRM_clip160.jpg118826":
{"filename":"33an2_SLRM_clip160.jpg",
"size":118826,
"regions":
        [{"shape_attributes":
                {"name":"polygon",
                 "all_points_x": [228,289,334,296],
                "all_points_y": [320,384,328,286]},
                "region_attributes": {"type":"celtic_field"}},
        {"shape_attributes":
                {"name":"polygon",
                "all_points_x": [242,281,348,316],
                "all_points_y": [219,279,241,192]},
                "region_attributes": {"type":"celtic_field"}},
        {"shape_attributes":
                {"name":"polygon",
                "all_points_x": [184,235,297,230],
                "all_points_y": [148,208,165,99]},
                "region_attributes": {"type":"celtic_field"}},
"file_attributes": {"caption":"","public_domain":"no","image_url":""}}
```


Figure 3.1: Annotations format example for a random image from the training set, containing three Celtic Field instances

Thus the data was annotated in the JSON format, more commonly used for Mask R-CNN training, using the VGG Image Annotator (VIA) (Dutta *et al.* 2016; Dutta and Zisserman 2019). This was done by creating a polygon to roughly outline the shape for each of the object 'instances', which in this case was individual Celtic Field plots, and adding a label to them defining them as such. The loc-

ations of the Celtic Field instances within the image are saved as pixel co-ordinates of the polygon outline drawn during annotation. The annotations were then compiled in a single JSON file, consisting of the image properties - *filename*, *size*, *shape attributes* including *name*, all_points_x and all_points_y (i.e. the pixel co-ordinates of the outlines), and *region attributes* defining the *type* as 'celtic_field'. Figure 3.2 shows the annotation process in the VIA tool, while Figure 3.1 shows an example of the annotations style for an image with id '33an2_SLRM_clip160.jpg118826', containing three instances of Celtic Fields.

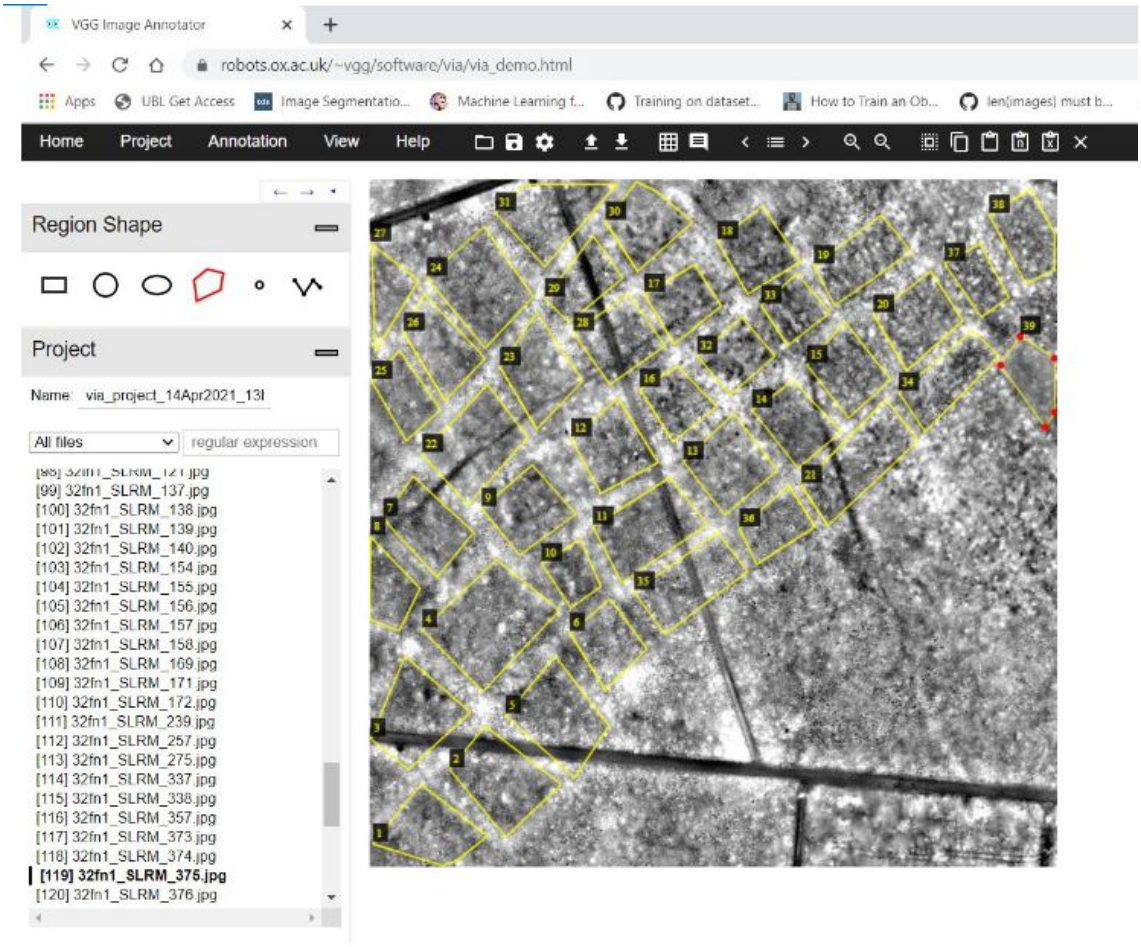


Figure 3.2: Data annotations done by drawing a polygon for each individual plot, using the VGG Image Annotator Tool

Figure 3.3 is a sample of the complete input data: four random input images from the training dataset (in the first column), along with the corresponding Celtic Fields ground truth mask for each, as

loaded with the annotations (second column). In case of a multi-class model, the ground truth masks for more objects would have been loaded in the rest of the columns. However since in this case we are only attempting to detect one archaeological object, the last two columns have been left blank.

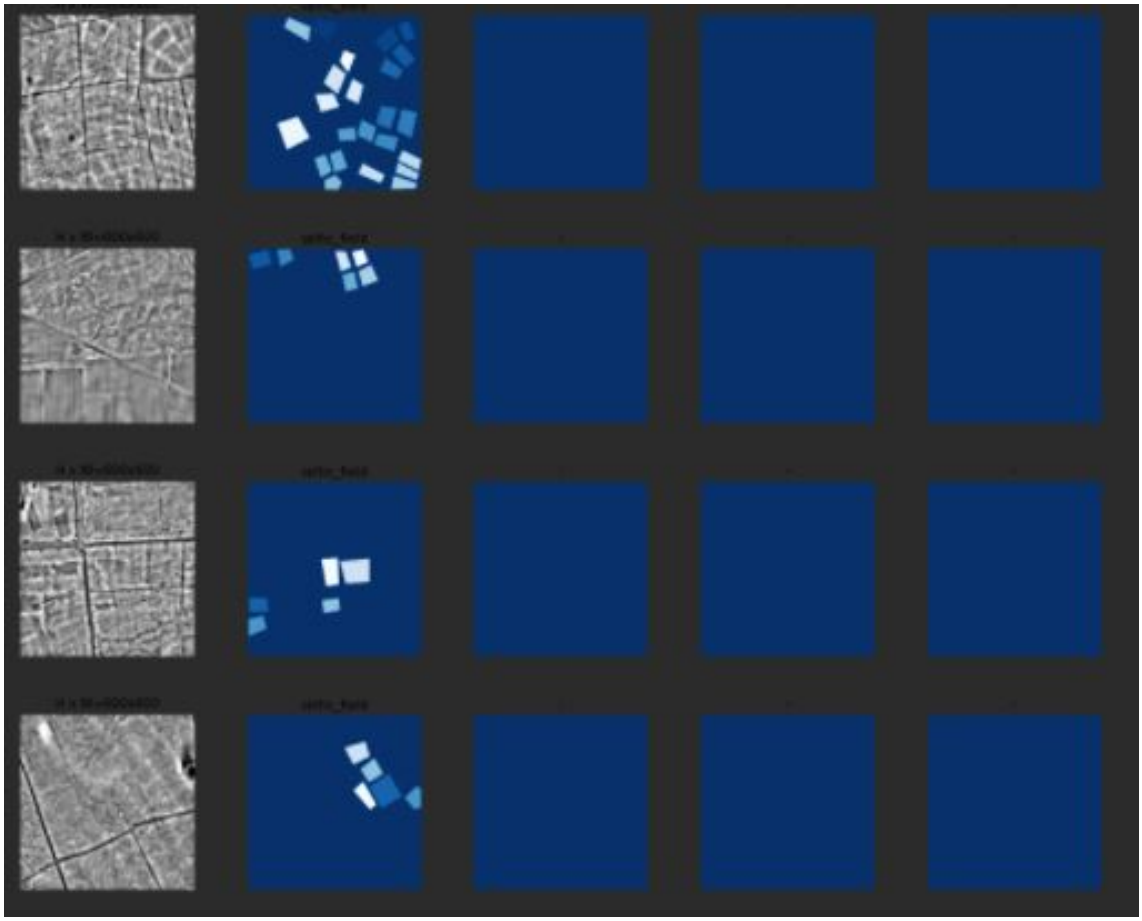


Figure 3.3: Sample images and the loaded 'masks' of the Celtic fields, from the training set

3.3 Training

The Mask RCNN algorithm used for the learning and inference process was the Matterport implementation script in Python (Abdulla 2017). This was altered and modified to fit the LiDAR dataset in this project.

Since the images were in greyscale format and the original implementation of Mask RCNN was for RGB images, two methods were available within the model implementation to work with greyscale

images. One was to convert all input images into RGB from greyscale, before training. Since the channel count is 3 for RGB and 1 for greyscale images, the information would be computed 3 times over. The other was to reduce the channel count itself to 1, and make changes in the code to accommodate the same. The data was trained and tested with both methods. Although it initially seemed that the second method would be better, since it would save more computational power, the first method of converting all input images into RGB provided better results after training, and was thus used for all experiments.

3.3.1 Transfer Learning

Deep neural networks as a rule are generally designed to work on significantly large datasets, for example thousands of images for computer vision tasks. However, in certain cases, a dataset of a sufficient size may be difficult to acquire. This is almost always the case for archaeology, as there are more often than not limited samples of a particular archaeological object or feature that can be used for training. To combat this limitation, a concept known as *transfer learning* is used (Pan and Yang 2010) for smaller datasets. In neural networks, the first few layers have been shown to extract more general features, while the last layers are more feature-specific (Yosinski *et al.* 2014, 1-3). The transfer learning approach involves pre-training a base network on a large generic dataset, to build up its learning abilities and then "transferring" the model weights to the first few layers of the CNN. The later layers are then 'fine-tuned' on the task dataset, to perform more specific training of features (Yosinski *et al.* 2014, 2), in this case for the Celtic Fields. The algorithm thus has a better chance of improving its detection capability, as opposed to training from scratch on a smaller dataset with insufficient training samples.

In certain cases, this fine-tuning process can lead to *overfitting*, especially on a small dataset. This is a form of modelling error, which fits the trained model too closely to a particular set of data. This reduces the ability of the model to generalise differences on new data and make future predictions. Therefore some feature layers can also be

frozen during the training process, to avoid the same (Yosinski *et al.* 2014, 2). This method has been proven to show improved generalisation capability of a deep neural network, as opposed to initialising the starting layers with randomised weights (Bengio *et al.* 2011; Yosinski *et al.* 2014).

The Matterport codebase provides pre-trained model weights for two widely used image sets: the Microsoft Common Objects in Context (COCO) (Lin *et al.* 2014) and ImageNet (Deng *et al.* 2009) datasets. MS COCO is a large scale benchmark dataset comprising of around 80 generic object classes. It is the basis of many computer vision based challenges, most notably instance segmentation (Lin *et al.* 2014). Experiments were conducted with both set of weights, however initialising with MS COCO showed better results.

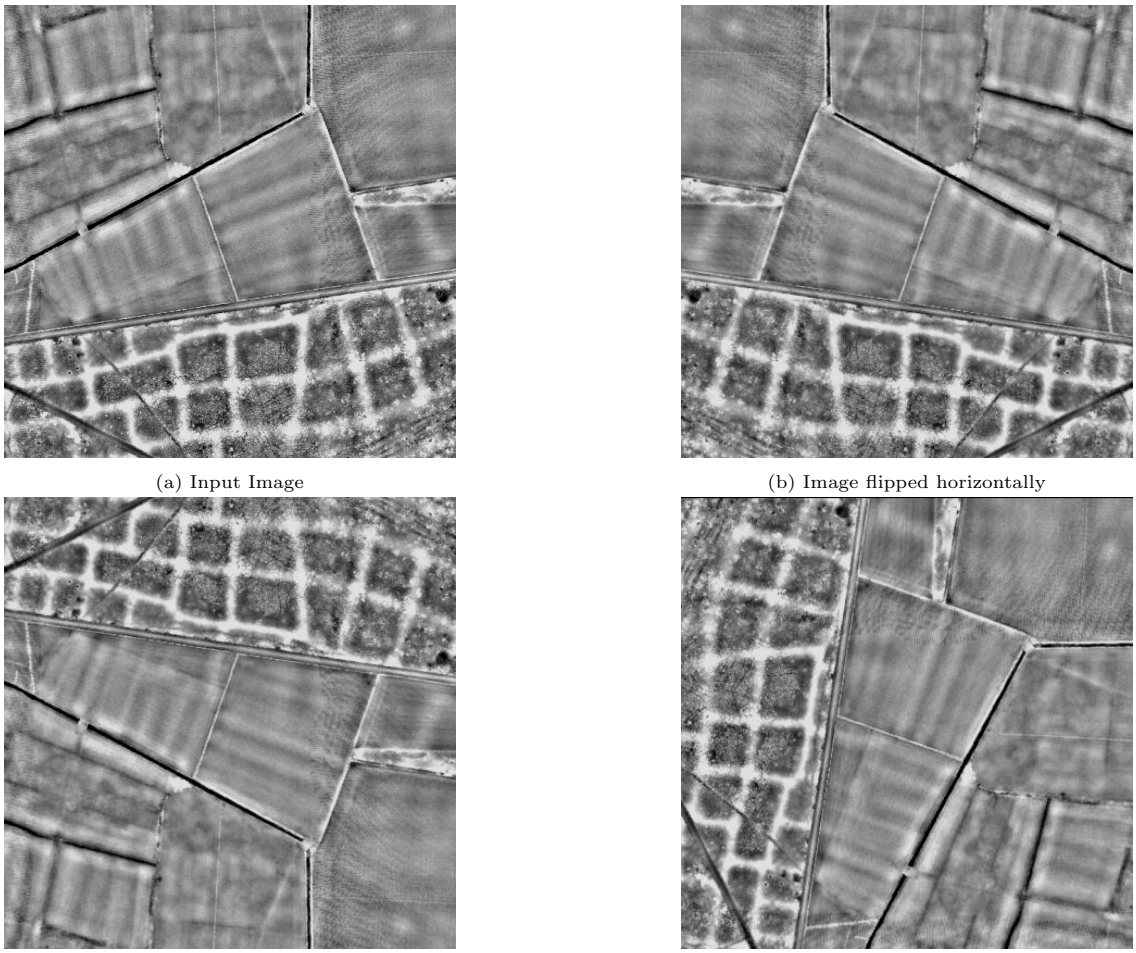
3.3.2 Image Augmentation

Another common method used to combat potential overfitting in smaller datasets is *image augmentation*. This is a method which can be used to increase the size of training data by generating new slightly modified images, using various image transformation techniques. Image augmentation techniques have also been shown to increase model accuracy in deep learning computer vision tasks (Perez and Wang 2017; Mikołajczyk and Grochowski 2018).

Since this project uses a relatively small training set, and thus lesser samples of Celtic Fields, initial experiments showed a large amount of overfitting, leading to poor level of generalisation on the validation set. Therefore, various data augmentation techniques were applied on the input data, every time before the model was trained. These were done using the *imgaug* library (Jung *et al.* 2020). The different types of augmentations experimented with were as follows:

1. *Fliplr* - Flipping the input image horizontally (applied to 50% of the data).
2. *Flipud* - Flipping the input image vertically (applied to 50% of data).
3. *Rotate* - Images were randomly rotated by either 90, 180 or 270 degrees

Figure 3.4 shows examples of the different augmentation techniques applied to an input image of the training set.



(a) Input Image

(b) Image flipped horizontally

(c) Image flipped vertically

(d) Image rotated by 90 degrees

Figure 3.4: Augmentations applied on input images before training

3.3.3 Experiments

The code implementation allows for training with both ResNet 50 & ResNet 101. The experiments in this project were conducted with the ResNet 101 backbone architecture as it has shown the best results with general Mask R-CNN implementations, as well past projects in archaeology as well (Bonhage *et al.* 2021; Kazimi *et al.* 2019).

In order to account for the type and size of the dataset, as well as the archaeological object, some steps were performed to adjust the hyperparameters from their default configurations (shown in Appendix

C), monitor and reduce overfitting and improve model accuracy.

- A validation step was plotted at the end of every epoch during training. At the end of the training process, a graph was plot showing the training and validation loss. More deviation between the two, with the validation loss being higher, hinted at overfitting. Accordingly the number of epochs were adjusted, by determining at which points the difference between the loss plots was increasing. This however was also cross checked at regular intervals against the actual predictions, since the discrepancy could also be explained away by factors other than overfitting, such as a lack of sufficient validation Celtic Field samples and thus proper generalisation of the model on the validation set.
- The RPN_Anchor_Scales were reduced from *(32, 64, 128, 256, 512)*, and varied between *(16, 32, 64, 128, 256)* and *(8, 16, 32, 64, 128)*. The former was too large relative to the size of the Celtic Fields.
- Number of training ROIs per image were reduced from 256 to 50, and maximum ground truth and detection instances were reduced from 100 to 40 each. This was done due to the relatively small number of objects present per input sub-image and to reduce training time.
- The number of training iterations (STEPS_PER_EPOCH) was set at 300. Initial experiments had been done with higher counts of 1000, 700 and 500, but this led to overfitting very early on due to the small size of the dataset.
- The L2 regularisation weight decay (another method used to reduce overfitting by reducing the weights at every iteration) was increased from 0.001 to 0.01. The model benefited greatly from this step, instantly improving performance.

Optimisers : While training a deep learning network, it is important to try and minimise the losses incurred during training, in order to maximise performance. This process is known as optimization. To do so, specific algorithms have been designed, which change and update

parameters, such as the weights and the learning rate, by minimising the loss functions. In this project, experiments were conducted using two commonly used optimisers for deep neural networks, namely - Stochastic Gradient Descent (SGD) with momentum (Robbins and Monro 1951; Qian 1999) and Adam (Kingma and Ba 2014).

Training Schedule & Learning Rates: Finding the model with an optimal combination of network configurations is a daunting task. This is because training the layers of a deep neural network is a time-consuming process, which can take anywhere from a few hours to a day/multiple days. Therefore, we can increase model efficiency by using a *training schedule*, to freeze some of the layers while training, and initialising them with the pre-trained transfer learning weights. The model was already hard-coded to do the same by specifying the layers to be trained: 'heads' refers to training only the network heads (RPN, FPN & Classifier), '3+'/'4+'/'5+' refers to training specific ResNet layers and above, and 'all' refers to training all layers in the Mask R-CNN architecture. The training schedules used for the experiment can be seen in Table 3.4 and Table 3.5. A decaying learning rate was used for the different stages of the training schedule i.e. it was reduced after every few epochs for better generalisation; the rate was higher in the first stage when the networks heads were fine tuned on the specific Celtic Field features, and lower in subsequent stages when more layers were being trained.

Table 3.4: Training schedule for experiments conducted using SGD optimizer

Training Stage	Layers Trained	Epochs	Learning Rate
1	*'heads'*	10	10^{-3}
2	*'all'*	20	10^{-4}
3	*'all'*	30	10^{-5}

Table 3.5: Training schedule for experiments conducted using Adam optimizer

Training Stage	Layers Trained	Epochs	Learning Rate
1	*'heads'*	10	10^{-4}
2	*'all'*	20	10^{-5}
3	*'all'*	30	10^{-6}

3.4 Inference

3.4.1 Evaluation Metrics

Evaluation metrics are mathematical formulas, which allow us to quantitatively assess the performance of the trained model on new data. There are certain common metrics which have been adapted for computer vision tasks. These calculations are mainly based on four classes of predicted values - *True Positive*, *False Positive*, *False Negative* and *True Negative*. These values can be plotted through a 'confusion matrix' on a test dataset (shown in Figure 3.5). As the names suggest, True Positives and True Negatives are positive and negative predictions that match the actual ground truth data (i.e. in this case a Celtic Field being detected or not detected where it is actually present/absent on the landscape) and False Positives and False Negatives are positive and negative predictions that do not match the ground truth reality.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 3.5: Confusion Matrix

For evaluation of deep learning neural network models, two primary metrics have been defined - *Precision* and *Recall* (Juba and Le 2019). 1. Precision: Refers to the ratio of relevant positive predictions (i.e. True Positives) out of total detected positive predictions. It is mathematically represented by Equation 3.1.

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

2. Recall: Refers to the fraction of positive predictions that have been

correctly identified with the respect to actual ground truth. Recall is a measure of sensitivity if a trained model. It is mathematically represented by Equation 3.3.

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

In order to define what constitutes a true positive prediction, a threshold value is defined, known as *Intersection Over Union* or IoU. It is defined as the intersection between the predicted and the ground truth mask, divided by their union. The commonly used threshold value is 0.5. Therefore, if the computed IoU > 0.5, the prediction is categorised as True Positive, else a False Positive.

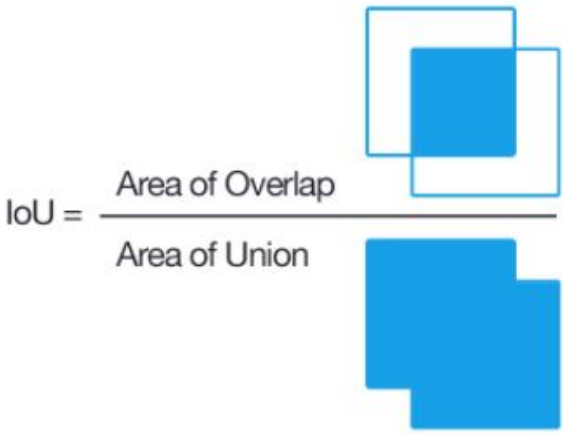


Figure 3.6: Representation of Intersection Over Union

3. *mAP*: For Mask RCNN challenges and implementations, the most widely used metric is the *Mean Average Precision* or *mAP*. The Average Precision (AP) value refers to area under a Precision-Recall curve, and the mAP is the mean AP over all object classes (in this case just the Celtic Fileds) at a certain IoU threshold.

$$mAP@IoU_x = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{3.3}$$

where x is the threshold value (in this case 0.5) and n is number of images in the given dataset.

4. *F1 Score*: The F1 score is a metric which represents the balance between precision and recall on the dataset. It is represented by the equation 3.4.

$$F1 = (2 * precision * recall)/(precision + recall) \tag{3.4}$$

3.4.2 Visualising Predictions

The final inference step consists of visualising the detected objects on unseen test images. The output image consists of: detections of the Celtic Fields, characterised by a class label, the predicted masks of the Celtic Fields which outlines their exact shape and finally a 'confidence score' i.e. a value between 0 to 1, which highlights the model's evaluation of how likely it is that it has detected a Celtic Field.

It is important to note that the results of this automated detection methodology is only part of a wider framework. It is important to further contextualise these results in a format which can aid archaeological research. To do so, the mask detection results can be exported as shapefiles to a GIS platform, and then observed with relation to the wider framework. Due to time constraints, this step was not conducted within this thesis. However a rough possible methodology for the same has been outlined below:

- To start with, the pixel coordinates of the generated mask predictions need to be written to a text file. One possible way to do so is by using OpenCV (a library developed for Computer Vision tasks) to create a 'convex hull' (https://docs.opencv.org/). This refers to creating an outline which mimics the outer shape of the object. The pixel values of this convex hull can then be written into '.txt' file, using a few lines of Python code.
- Subsequently the convex hull co-ordinates need to be converted to real world geographic co-ordinates, in order to read it in a GIS format. This can be done using Python's GDAL package (https://gdal.org/), built for manipulating geospatial raster data.

This can then be opened as a shapefile in QGIS, and multiple larger Celtic Fields (made up of the individual small plots detected in this method) can be visualised on the wider landscape of the Veluwe region

Chapter 4

Results

The following section comprises of results from the experiments (as described in the previous section), consisting of quantitative metric values derived from each experiment, as well as visualised predictions of some sample images from the test set (not used during training).

In total, almost 30-40 experiments were conducted with different combinations of hyperparameters, training schedules and optimisers. Of these, only the most significant results are present, in terms of most optimal performance. The results of the different experiments were evaluated using the Mean Average Precision (mAP), precision, recall and F1 score, for an IoU of 0.5, and minimum detection threshold of 0.7. This value was set in order to prevent the detection of false positives as much as possible. Of these metrics, the mAP was considered the most significant, in keeping with most Mask RCNN implementations, while F1 score was used for comparison with other results achieved in the Faster R-CNN project.

4.1 SGD Optimiser

Table 4.1 shows the results for the model trained with the SGD optimiser, with the configurations and training schedule mentioned in Sec 4.5.2 and Table 3.4.

Table 4.1: Results of experiments conducted with varying anchor scales; best performing model is experiment 2 with an mAP of 0.53 and F1 score of 0.60.

Experiment	Anchor Scales	Weight Decay	mAP	Precision	Recall	F1
1	(8, 16, 32, 64, 128)	0.01	0.491	0.75	0.46	0.57
2	(16, 32, 64, 128, 256)	0.01	**0.53**	0.77	0.50	0.60

Out of the two, we see that the second experiment, with anchor scales *(16, 32, 64, 128, 256)*, has higher metrics on the test set, with an mAP of 0.53. We can presume that this difference is due to the anchor scales size of the first experiment being too small to effectively cover the area of the Celtic Field plots. Visual inference with the second experiment showed a good coverage of Celtic Field plots, however there were some overlaps in the masks. This was probably because some of the LiDAR images had blurred sections within them, which might prevent a proper distinction between the embankments and the actual plots.

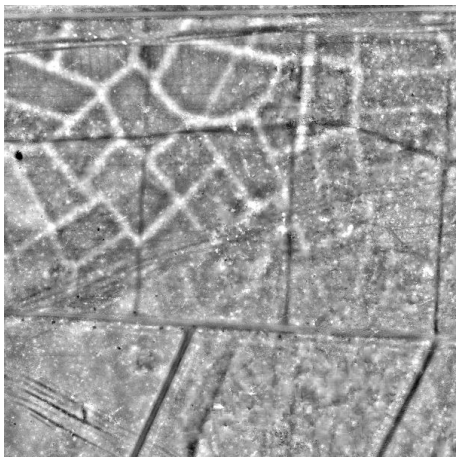
While these instances of overlapping masks were largely negligible, another experiment was conducted with the same anchor scales, but by slightly altering the Mask Loss weights, to try and improve accuracy of the mask delineation. In addition, a fourth experiment was conducted using a step decay function, wherein the learning rate dropped by 10 after every 10 epochs. This was done in order to further gauge the effect of the learning rate on the model.

Experiment	Configuration	mAP	Precision	Recall	F1
3	Altered loss weight	0.456	0.65	0.35	0.455
4	Step Decay	0.51	0.63	0.267	0.38

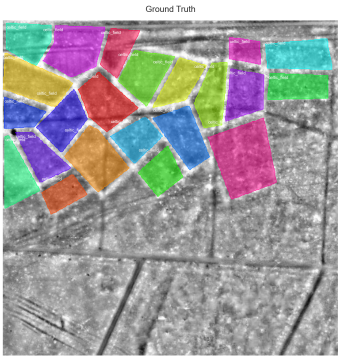
Table 4.2: Results of experiments conducted with altered configurations

Experiments 3 and 4 show reduced metric values rather than improvement. Thus, Experiment 2 remains the best performing model. Figure 4.1 and Figure 4.2 show predicted results on two images from

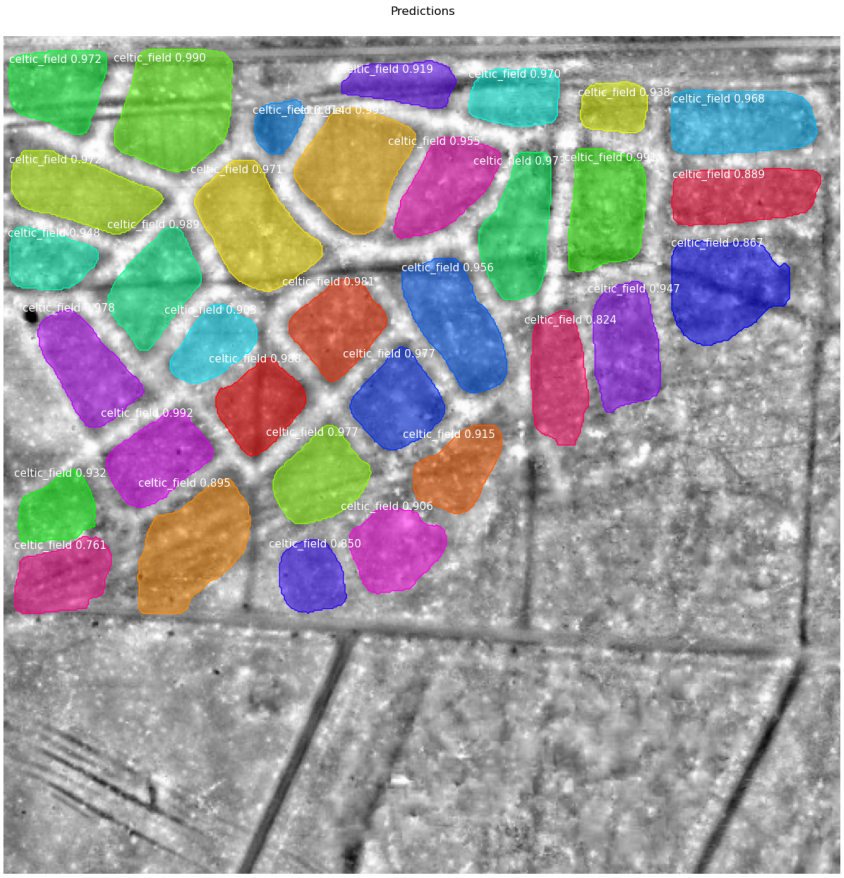
the test dataset, along with the original image and the ground truth as generated by annotations. The annotations in this case were only fed into the algorithm in order to compare the results with the final results, and do not play a part in the training process. In both cases, we see that some extra fields, not uploaded in the ground truth have also been detected. In 4.2c, we can see an example of a false positive in the centre right part of the image (marked in green).



(a) Input Image

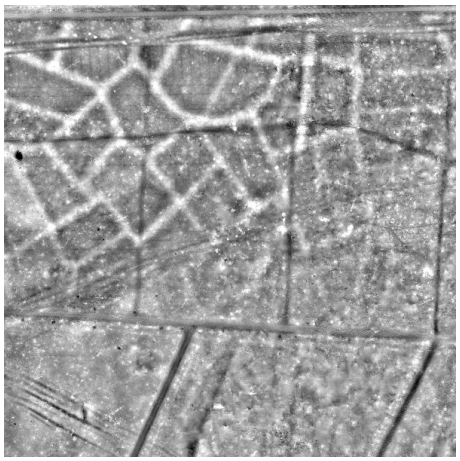


(b) Ground Truth

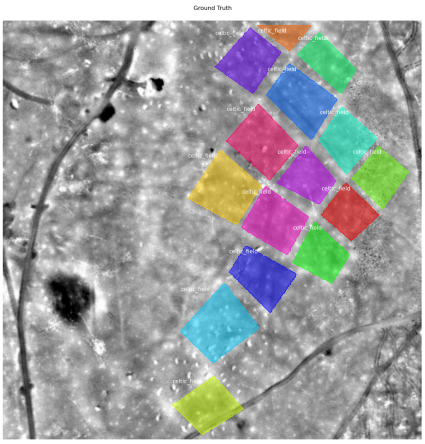


(c) Predictions

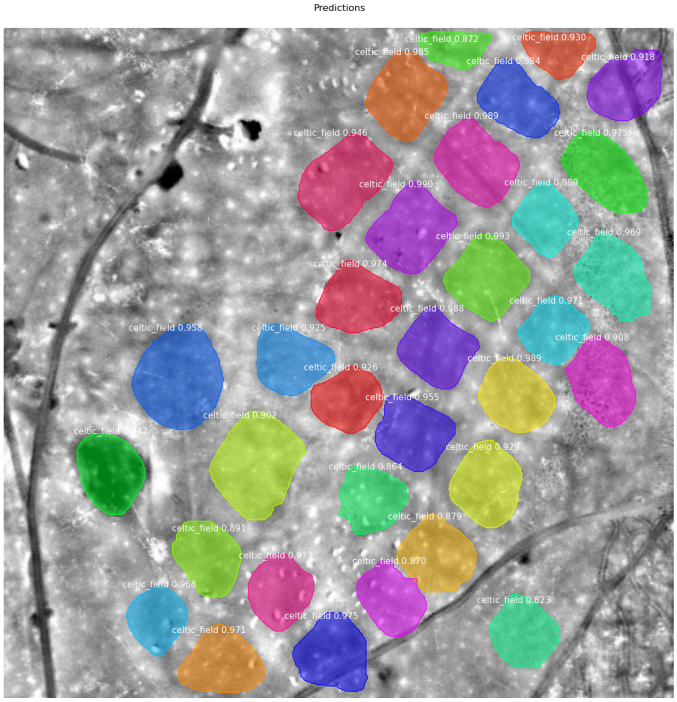
Figure 4.1: Predictions on first example (image from test set). We see that some instances not marked in the original annotations have also been detected



(a) Input Image



(b) Ground Truth



(c) Predictions

Figure 4.2: Predictions on second example (image from test set)

4.2 ADAM Optimiser

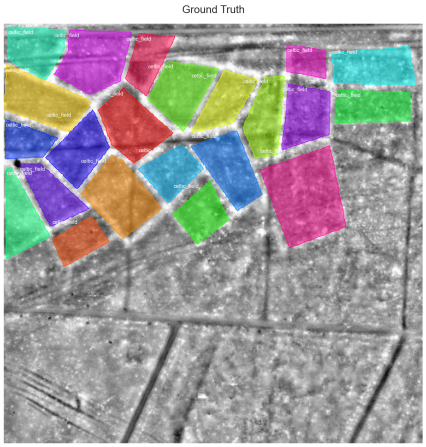
The next set of experiments was conducted using the Adam Optimiser. For this the decay rate of the optimiser was varied, along with the number of RPN Training anchors generated.

Table 4.3: Results of experiments conducted with altered configurations

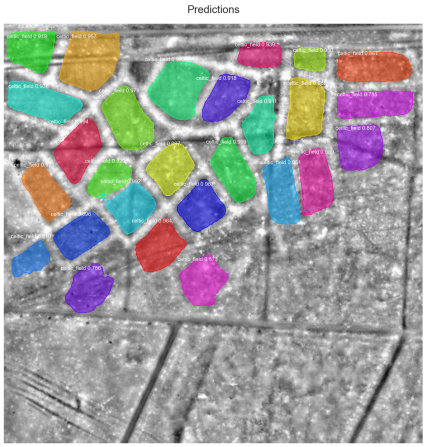
Experiment	Decay	Training anchors	mAP	Precision	Recall	F1
5	10^{-5}	128	0.436	0.57	0.48	0.52
6	0	64	0.451	0.62	0.40	0.48

Since the results were significantly lower then that of the SGD optimiser, further experiments were not conducted. However, the amount of time taken for training and loss convergence were considerably lower than with the SGD optimiser, for the same number of iterations and epochs.

With respect to visualised predictions as well, the Adam optimiser model was able to predict fewer instances of Celtic Fields than the SGD optimiser. Predictions on the two example images can be seen in Figure 4.4 and 4.3. These are the same examples as used for the SGD optimiser. In addition, the contour mask as well was not as well defined as in the case of the SGD optimiser. This can be seen particularly in Figure 4.4 in the top right corner, for the fields depicted in green and pink. Aside from the overlap. there are also fragmented smaller masks separate from the actual Celtic Field plot. This hows a lower generalisation of the model in determining the exact shape of the fields, as compared to experiments in the previous section.

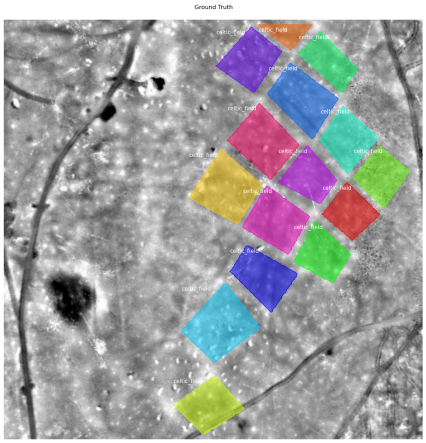


(a) Ground Truth

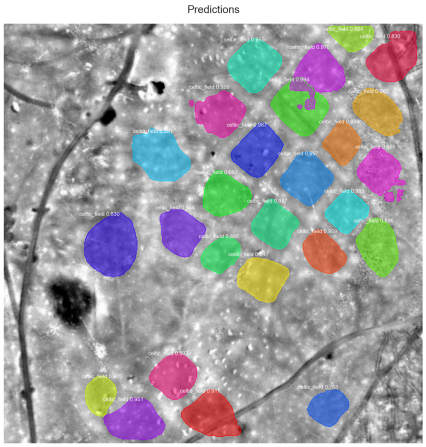


(b) Prediction

Figure 4.3: Predictions on first example (image from test set)



(a) Ground Truth



(b) Prediction

Figure 4.4: Predictions on second example (image from test set)

4.3 Summary

- The best performing model is Experiment 2, using the SGD optimiser, with a weight decay of 0.01 and Anchor Scales (16, 32, 64, 128, 256). The mAP value achieved was 0.53, with a recall and precision of 0.50 and 0.77 respectively. In terms of predictions, the model was able to detect a majority of the plots within the larger Celtic Field systems and generalised the shape quite well.
- Experiments conducted using the SGD optimiser gave better results as compared to the ADAM optimiser. However, the latter required a lesser amount of training time.
- A parameter which caused a significant change in model performance was the weight decay rate for the L2 optimisation. The default value was 0.0001. This was then increased to first 0.005 and then finally 0.01. The value increased the metrics from as much as 0.3 to 0.5.
- Since the input images in the dataset are smaller slices of the larger LiDAR images of the region, integrating the results into GIS and visualising the predictions on the landscape would give a clearer idea of the position of larger Celtic Field systems in the region.
- Additional Comment: The randomised color scheme of the predictions is built into the original Mask R-CNN implementation, and does not in this case give any additional information about the masks created.

Chapter 5

Discussion

5.1 Input Data

The format and quality of input data and the annotations used for training is very important in this methodology as the relevant features (for identification) of the celtic Fields are learned by the algorithm itself, from the annotated images, rather than being fed manually by the programmer. This is also exemplified when looking at the predictions generated when using the previous annotations (Appendix D).

Future research could include assessing the performance of this model on a different format of the input data. Verschoof-van der Vaart and Landauer (2021) in their research on using CNNs to detect hollow roads in the Veluwe note that the model performed better when the input data was in raw DTM format, as opposed to visualised LiDAR data (using hillshade). This was attributed to a loss in data information when converting from one format to another (DTM to visualised) (Verschoof-van der Vaart and Landauer 2021). Similarly, the model could also be tested against other forms of LiDAR visualisations to see what impact they have on performance.

5.2 Model Evaluation

The results of the different experiments and the resulting predictions were evaluated using the Mean Average Precision Metric (mAP), for an IoU of 0.5, and minimum detection threshold of 0.7. This value was set in order to prevent the detection of false positives as much

as possible. Different variations of the experiments were applied, by varying the anchor scales and adding a step decay rate as opposed to a constant learning rate. These changes had little impact on the mAP value. The Adam optimiser was also applied as it it is touted to have a better convergence of the learning loss and also greater accuracy. However it performed less well as compared to the SGD optimiser. The best model results, in quantitative terms, was Experiment 2, using the SGD optimiser, with a weight decay of 0.01 and Anchor Scales (16, 32, 64, 128, 256). The **mAP value** achieved was **0.53**, with a recall and precision of **0.50** and **0.77** respectively.

To the best of this author's knowledge, there have been no other projects which have attempted the automated detection of Celtic Fields specifically from any kind of remote sensing data. The metrics therefore were compared only to the results of the original object detection methodology used on this data. The comparison was done on the basis of precision, recall and F1 scores.

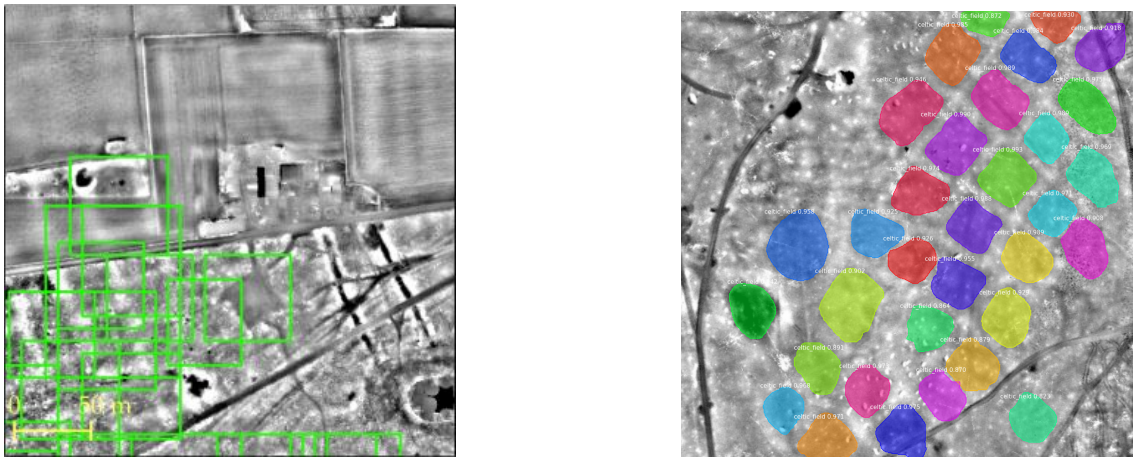
Methodology	Precision	Recall	F1 sco
WODAN 1.0 (Verschoof-van der Vaart and Lambers 2019)	57.6	82.3	67.8
WODAN 2.0 (Verschoof-van der Vaart *et al.* 2020)	66.0	74.6	70.0
Heritage Quest:Citizen Science (Verschoof-van der Vaart *et al.* 2020)	85.0	75.7	80.1
Mask R-CNN	77.0	50.0	60.0

Table 5.1: Comparison of metric performance with previous models

The precision values of the current model are higher than the previous two implementations of WODAN, using the object detection methodology, whereas the recall is much lower. The latter is probably the reason why the mAP and F1 scores across all experiments are also lower, as these metrics take into account the trade-off between the precision and recall. Precision gives an idea about the amount of true positives versus false positives predicted, hence it can be considered more significant for the purposes of archaeological prospection. The model does not however reach the level of the results of the citizen science project i.e. human performance.

One of the research aims defined was assessing the performance of an instance segmentation based model (Mask R-CNN) versus object detection (Faster R-CNN) in the detection of Celtic Fields. Results

from both methodologies can be seen in Figure. In terms of actual detection capability of the field instances, the results are comparable as both seem to positively identify most of the field plots in an image. However we see a difference in the representations of the detection. The object detection method covers the general field plot with a bounding box. Due to the nature of this particular archaeological object, wherein the fields are all present adjacent to each other in the landscape, the bounding boxes have a great amount of overlap, thus causing some confusion about the actual positioning of the individual field plots. The Mask R-CNN implementation on the other hand clearly delineates the instances from each other and gives a relatively good real world view at how the Celtic Fields are positioned on the landscape. Thus, in terms of archaeological prospection, this method seems to be better for detection of individual elements of a larger archaeological system, especially which have little to no spatial separation between them. Another limitation of the bounding boxes is that they have a very static rectangular shape. This is why they work well for uniformly shaped objects such as barrows and charcoal kilns that would fit well within these boxes and be highlighted within the image. In case of Celtic Fields however, which have non uniform shapes that do not fit into these boxes, generalising the exact contours provides a clearer and more coherent picture.



(a) Object Detection Results

(b) Instance Segmentation Results

Figure 5.1: Predictions using the two methodologies

5.3 Archaeological Significance

While the various model metrics provide an efficient quantitative way to evaluate the model performance, it is important to look at the qualitative value of the methodology from an archaeological perspective. The mAP metric achieved so far i.e. 53% would not be considered very high in circle of Deep Learning research, but considering it by itself as a measure of the model's usefulness for the specific task in hand would be premature. Since Celtic Fields are a larger system made up of a combination of smaller 'objects' (i.e. smaller, individual field plots), successfully outlining the core of the fields would allow a reconstruction of a few of the plots that might have been missed out in the original model detection. In addition, the methodology needs to be embedded into a wider framework, so that the results can be used for archaeological research. This could include elements such as manually checking the predicted results and analysing, ground truthing the model results through through field survey and finally visualising the results on the larger landscape and asking interpretations. Another step that could be incorporating domain knowledge into the results, such as the use of Location Based Ranking by Verschoof-van der Vaart *et al.* 2020 to account for effect of present landscape conditions on the prediction results.

Currently, the model provides quite a clear delineation and mapping of individual field plots, within the 'walls' or demarcations inside the wider Celtic field. Figure **??** gives a rough look at what multiple field systems on the landscape imagery might look like. Further analysis of this can help answer important archaeological questions regarding land use, agricultural systems and past human-agricultural dynamics in the research region. Arnoldussen (2018) in his research covers the different ways in which Celtic Fields need to be studied from an archaeological point of view. This includes making inferences about the adaptability of agricultural structures by studying the morphology of the fields across regions (Arnoldussen 2018, 7). This can easily by done through the results of this methodology, by extracting characteristics such as morphology, area and perimeter of these embanked fields, using GIS. A similar approach was followed by Guyot *et al.*

(2021). Moreover, further analysis could be made about the size of settlements with relation to area of agricultural land, thus providing a non-destructive methodology to study habitation-agriculture relations in the research area (Arnoldussen 2018, 9). Statistical analysis techniques can be used to determine spread and size of the wider field systems with respect to the different municipalities in the Veluwe region, and make inferences and correlations.

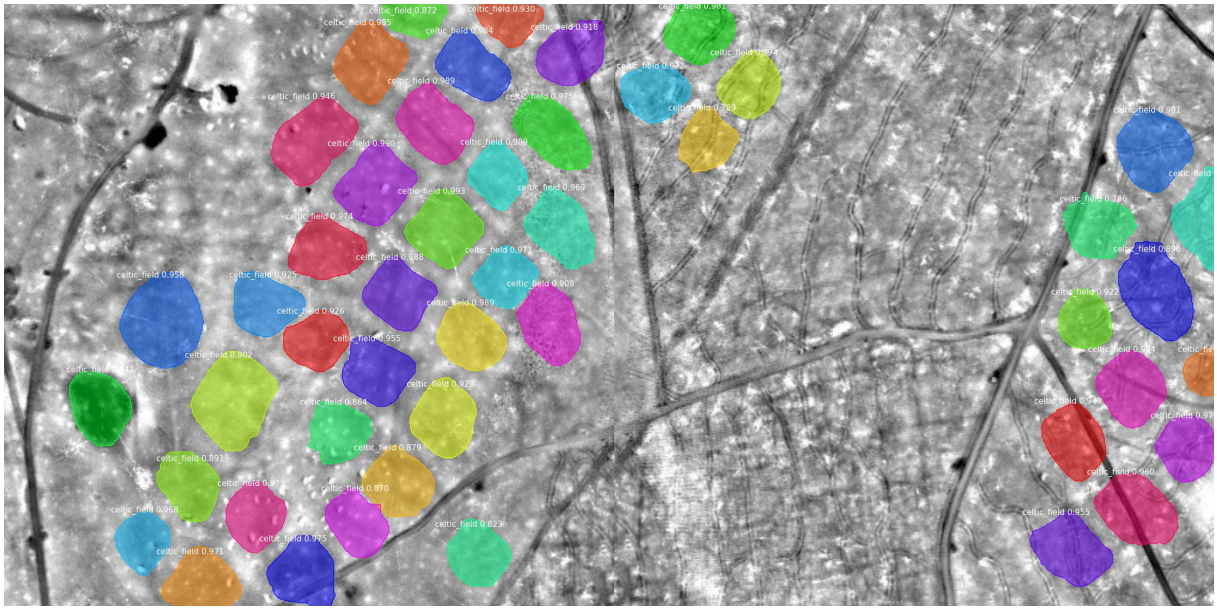


Figure 5.2: Two merged images from the test set, to try and reconstruct a small part of the landscape.

A point of consideration is that the model, as it stands now, does not include the 'banks' or walls around the plots. The banks themselves contain important archaeological residue and settlement debris such as firewood, charcoal, sherds etc (Arnoldussen 2018, 313). Moreover, studying their formation processes gives important insight into the land-use and agricultural history. Detecting these could also help in further quantitative analysis, such as calculation of perimeter and area of the wider field systems. There could be a number of ways to achieve this. One method would be adjusting the annotations and the model parameters, and retraining with the data, in order to incorporate the banks of the fields in the predictions. However this would obviously be a more computational and cost-inefficient method. Moreover, the current model shows very clear delineations between

the fields, in the context of the landscape, and that might not be the case with a new model. Instead, post-processing techniques could be applied to the current results. One way to do this would be to import the predictions to GIS, and use some of the tools to try and outline the field embankments. An example is the buffer analysis tool (https://docs.qgis.org/2.8/en/docs/training_manual/).

When studying the model, another way to recontextualise the results is in terms of the targeted application. The precision and recall values of a model has a trade-off, such that maximising one, minimises the other. If the goal is archaeological prospection and analysis, greater precision would be required of the model. That is, it would be important to eliminate as many of the false positives as possible. It would also be important to have as much accuracy as possible in terms of outlining the shape of the fields, as this would reduce post-processing steps when for example analysing morphometric characteristics. If the aim however was along the lines of heritage management and conservation, it would be more important to maximise the recall detect as many possible occurrences of these fields as possible. The number of false positives in this case would not matter so much, rather higher number of false negatives would mean missing out on existence of archaeologically relevant area that might require protection. In this way, 'improving the model' might require very different techniques, based on the end goal. Moreover, since we get a complete look at the statistics of detected and undetected features, it is possible to calibrate the results in the necessary way.

5.4 Considerations & Problems of the methodology

Input Data: While an obvious point of consideration, the duration of this project exemplified the impact the quality and format of input data could have on the final results. Initially, experiments and model training and inference were conducted using the original bounding box annotations as used for detections using Faster R-CNN. While the actual positive detections were comparable, the masks created were less accurate, encompassing a much larger area than the actual field plots, and causing incoherence in the visualisation of the predictions.

Another limitation is that, in order to make the input data feasible for the computational capabilities of the neural network, the data had to be sliced into smaller sets. Thus, individual predictions that we get from a test set image lack in giving contextual information with regards to the entire landscape. To mitigate this, an extra step is required by exporting all of the predicted data into a GIS environment, and observing the individual 'slices' as part of the landscape of the research area.

Computational power: As mentioned in the requirements section as well, the methodology implemented requires a significant amount of hardware-based computational power, which is not so easily available on common PCs. The experiments of this project were conducted using a GPU system with a capacity of 4GB. On occasion, this did lead to the program running out of allocation memory, requiring a restart of the training process, sometimes from the last working point, but sometimes also from the start. Moreover, running the program on a device not having the sufficient amount of required memory, would lead to a longer training process and less accurate results.

Computational expertise: The entire workflow used in this methodology consisted of a number of processes which required different software platforms. This includes the annotation process, the actual ML algorithm processing and any post-processing techniques, if applied. Thus, the workflow as a whole lacks an element of ease-of-use, in terms of having to use different platforms for the task. Moreover, there is a strong coding component involved as well. Thus, the current methodology applied could be made more useful, if all of these stages i.e. pre-processing, the ML algorithm and post-processing could be integrated into a single platform. This could possibly be done on the Python IDE, since there are packages in the Python documentation which could allow the creation of annotation tools, as well as interfacing with GIS.

Interpreting results: The article by Roccetti *et al.* (2020) on the use of CNN in the Mesopotamian Floodplain sheds some light into how it would be imprudent to simply rely on the results of a CNN algorithm, without proper post interpretive analysis. The research shows that model accuracy and detection capability were improved

by - a) modification of the original model by adding background contextual information as input and b) using prediction heat-maps to interpret & improve the results (Roccetti *et al.* 2020). It was the second method, involving a combination of domain-based knowledge with initial model results, that improved accuracy more significantly, over simply adding tweaks to the original model (Roccetti *et al.* 2020, 4-5). Similar results were also seen in the use of Location Based Ranking, incorporating geomorphological and topographical data to the original detection workflow for data from the Veluwe, by Verschoof-van der Vaart *et al.* (2020). Therefore the use of Deep learning cannot be categorised as a "one fits all" solution, rather it requires an in depth interpretive analysis of the results and uncertainties from an archaeological point of view.

Chapter 6

Conclusions

Automated detection of archaeological objects from remote sensing data has gained much prominence in the recent decade. This occurrence is due to the constraints of time and cost inefficiency relating to the manual interpretation of large amounts of varied remote sensing data available in the present. Past implementations generally relate to handcrafted and rule-based methodologies, which are limited in their usability and transferability. A recent solution to this limitation is the use of Deep Learning-based CNN techniques. The aim of this thesis was to use such an approach to create a model that can perform instance segmentation of Celtic Fields, from LIDAR data collected from the Veluwe, Netherlands. The expected results were two-fold: a) the detection, localisation and labelling of Celtic Fields and b) delineating between individual plots in the wider field system, by generalising the exact shapes of individual plots. This was done using the widely used state-of-the-art instance segmentation architecture known as Mask RCNN. In order to account for the problem of the small size of the dataset, the methodology made use of transfer learning using the large scale dataset MS COCO.

A number of different experiments were conducted in order to find the model with optimum performance. This was done by fine-tuning the model using different combinations of hyper-parameters, such as learning rate, training schedules and so on. The model performance was subsequently validated through model metrics, specifically the mean average precision value, and the visualised predictions of the detection, with respect to the ground truth. The best model metric obtained was an mAP value of 53%. The methodology covered in this

project can be considered a part of a wider framework required to do a complete archaeological interpretation of the positioning of Celtic Field systems in the wider landscape, and subsequently assess the role they played in the region.

6.1 Research Questions

At the start of the thesis, a research question was defined - *How does a Deep Learning based Mask R-CNN algorithm perform with respect to the instance segmentation of pre-historic Celtic Fields, from remotely sensed LIDAR data of the Netherlands and subsequently contribute to archaeological prospection?*.

To answer this question, a number of sub-questions were defined. These questions have been answered in below.

1. To what extent can the model identify instances of individual Celtic Field plots and delineate between them?

The model has a high precision rate of 0.77. This combined with the visualised predictions indicates the model does a decent job at eliminating false positives within the predicted results. However the recall rate is low, which means that the total number of predictions in comparison to the ground truth is still possibly on a slightly lower side. However, with reference to Celtic Fields, it is not absolutely necessary to detect every single field plot. This is because Celtic Fields are a wider network of individual plots making up a bigger field system As a result, detecting most of the core of the fields can lead to a reconstruction of some of the instances which may have been missed out, depending on the necessity of doing so. The visualised predictions from the model show a clear delineation of the predicted fields, and covers of the wider field system, allowing a further reconstruction if necessary.

2. In what way do the results compare to a comparative object detection approach, previously applied on the data of the region?

In terms of the quantitative metrics used to evaluate the model, the results are comparable to the previous versions of object detection implementations. The Mask R-CNN model shows a higher precision rate, but a lower recall and subsequently F1 score. They however

do not match to the level of human performance as seen in the citizen science experiment which was conducted as part of the original project.

In terms of visualised predictions however, a difference can be seen with respect to the Mask R-CNN and object detection methodologies. The former shows more coherence and clarity in terms of the visualised predictions, delineating each individual field plot very clearly. In addition, it also outlines the contours of the fields quite significantly, which leaves open the option of applying further morphometric analysis with fewer post-processing steps.

3. How can the results obtained from the methodology contribute to archaeological interpretations of Celtic Fields in the Veluwe region?

This methodology can help in finding and detecting new Celtic Fields in a quick and efficient manner. Subsequently, when these predictions made on smaller input images are exported to GIS, they can be mapped on the larger landscape of the Veluwe. By combining these results with other sources, such as written, geographical and topographical data, we can better understand the role of Celtic Field systems in the region, in terms of human activity. We could also assess the relationship between these field systems, and other archaeological objects in the region. In addition, we can calculate quantitative measures of these fields, such as area and perimeter. The method can also be tweaked further for purposes of both archaeological prospection and heritage management by reducing false positives/true positives.

6.2 Future Work

An important aspect which can be tested in the future is the ability of the methodology to generalise on Celtic fields in the landscape from other parts of north-western Europe. This can give an idea of how transferable the current methodology is to other comparable situations, and to what extent this can reduce the invariable computation costs of re-training Deep Learning algorithms for detection of the same type of object from different data.

An important factor observed in the duration of this project is the lack of resources and examples from which to draw reference from,

with respect to the technical and methodological aspects. This is due to a vast underutilisation of platforms such as GitHub, which would allow the sharing of programming codes and workflows, large scale general datasets more appropriate for archaeological remote sensing applications and hypertuning and optimisation methods. This is specially important in arhcaeology due to the specific problems faced in the domain with respect to use of AI methods, such as lack of large training datasets and samples and uncertainty within the data. A strong community on such platforms would facilitate the process of research into this domain for future digital archaeologists.

Webpages

(1) https://www.usgs.gov/faqs/what-remote-sensing-and-what-it-used?, accessed on 13 March 2021
(2) Publieke Dienstverlening Op de Kaart (PDOK). https://www.pdok.nl/, accessed on 23 March 2021
(3) Actueel Hoogtebestand Nederland (AHN). ahn.arcgisonline.nl/ahnviewer/, accessed on 23 March 2021
(4) https://pypi.org/project/labelImg/1.4.0/ (accessed on 23 March 2021)
(5) Python resources and documentation. https://www.python.org/, accessed on 23 March 2021
(6) https://deepai.org/machine-learning-glossary-and-terms/computer-vision, accessed on 29 March 2021
(7)https://www.tensorflow.org/tutorials/images/segmentation, accessed on 29 March 2021
(8)https://analyticsindiamag.com/semantic-vs-instance-vs-panoptic-which-image-segmentation-technique-to-choose/, accessed on 5 April 2021
(9)https://www.jetbrains.com/pycharm/, accessed on 5 April 2021
(10)https://docs.qgis.org/2.8/en/docs/training_manual/vector_analysis/spatial_sta accessed on 5 May 2021
(11)https://docs.opencv.org/3.4/d7/d1d/tutorial_hull.html, accessed on 16 June 2021
(12)https://gdal.org/api/python.html, accessed on 16 June 2021

Bibliography

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zhang, 2016. Tensorflow: A system for large-scale machine learning. *arXiv preprint* .

Abdulla, W., 2017. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN.

Adelson, E., C. Anderson, J. Bergen, P. Burt, and J. Ogden, 1983. Pyramid methods in image processing. *RCA Eng.* 29.

Agapiou, A., D. D. Alexakis, and D. G. Hadjimitsis, 2014. Spectral sensitivity of alos, aster, ikonos, landsat and spot satellite imagery intended for the detection of archaeological crop marks. *International Journal of Digital Earth* 7(5), 351–372, doi:10.1080/17538947.2012.674159, URL https://doi.org/10.1080/17538947.2012.674159.

Alpaydin, E., 2014. *Introduction to machine learning*. Adaptive computation and machine learning, MIT: The MIT Press.

Aqdus, S., W. Hanson, and J. Drummond, 2012. The potential of hyperspectral and multi-spectral imagery to enhance archaeological cropmark detection: A comparative study. *Journal of Archaeological Science* 39, 1915–1924, doi:10.1016/j.jas.2012.01.034.

Arnoldussen, S., 2018. The Fields that Outlived the Celts: The Use-histories of Later Prehistoric Field Systems (Celtic Fields or Raatakkers) in the Netherlands 84, 303–327, doi:10.1017/ppr.2018.5.

Bakker, M. A. and J. Meer, 2003. Structure of a pleistocene push moraine revealed by gpr: the eastern veluwe ridge, the netherlands. *Geological Society, London, Special Publications* 211, 143–151, doi: 10.1144/GSL.SP.2001.211.01.12.

Bengio, Y., F. Bastien, A. Bergeron, N. Boulanger–Lewandowski, T. Breuel, Y. Chherawala, M. Cisse, M. Côté, D. Erhan, J. Eustache, X. Glorot, X. Muller, S. Pannetier Lebeuf, R. Pascanu, S. Rifai, F. Savard, and G. Sicard, 2011. Deep learners benefit more from out-of-distribution examples. In G. Gordon, D. Dunson, and M. Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, *Proceedings of Machine Learning Research*, vol. 15, 164–172.

Bennett, R., K. Welham, R. A. Hill, and A. L. J. Ford, 2012. The application of vegetation indices for the prospection of archaeological features in grass-dominated environments. *Archaeological Prospection* 19(3), 209–218, doi:https://doi.org/10.1002/arp.1429.

Bevan, A., 2015. The data deluge. *Antiquity* 89, 1473–1484, doi:10.15184/aqy.2015.102.

Bickler, S. H., 2021. Machine learning arrives in archaeology. *Advances in Archaeological Practice* 9(2), 186–191, doi:10.1017/aap.2021.6.

Boer, A. d., 2007. Using pattern recognition to search lidar data for archeological sites. In A. Figueiredo and G. L. Velho (eds.), *The world is in your eyes. CAA2005. Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 33rd Conference*, 245–254, Tomar.

Bonhage, A., M. Eltaher, T. Raab, M. Breuß, A. Raab, and A. Schneider, 2021. A modified mask region-based convolutional neural network approach for the automated detection of archaeological sites on high-resolution light detection and ranging-derived digital elevation models in the north german lowland. *Archaeological Prospection* 1–10, doi:https://doi.org/10.1002/arp.1806.

Bourgeois, Q., 2013. *Monuments on the Horizon: The Formation of the Barrow Landscape throughout the 3rd and 2nd Millennium BC*. Leiden.

Bundzel, M., M. Jaščur, M. Kováč, T. Lieskovský, P. Sinčák, and T. Tkáčik, 2020. Semantic segmentation of airborne lidar data in maya archaeology. *Remote Sensing* 12(22), 3685.

Caspari, G. and P. Crespo, 2019. Convolutional neural networks for archaeological site detection – Finding "princely" tombs. *Journal of Archaeological Science* 110, 104998, doi:10.1016/j.jas.2019.104998.

Cerrillo-Cuenca, E., 2017. An approach to the automatic surveying of prehistoric barrows through lidar. *Quaternary International* 435, 135–145, doi:https://doi.org/10.1016/j.quaint.2015.12.099.

Chase, A. F., D. Z. Chase, C. T. Fisher, S. J. Leisz, and J. F. Weishampel, 2012. Geospatial revolution and remote sensing LiDAR in Mesoamerican archaeology. *Proceedings of the National Academy of Sciences of the United States of America* 109(32), 12916–12921, doi:10.1073/pnas.1205198109.

Chen, F., J. You, P. Tang, W. Zhou, N. Masini, and R. Lasaponara, 2018. Unique performance of spaceborne sar remote sensing in cultural heritage applications: Overviews and perspectives. *Archaeological Prospection* 25(1), 71–79, doi:https://doi.org/10.1002/arp.1591.

Cheng, G. and J. Han, 2016. A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing* 117, 11–28, doi:https://doi.org/10.1016/j.isprsjprs.2016.03.014.

Coluzzi, R., A. Lanorte, and R. Lasaponara, 2010. On the LiDAR contribution for landscape archaeology and palaeoenvironmental studies: The case study of Bosco dell'Incoronata (Southern Italy). *Advances in Geosciences* 24, 125–132, doi:10.5194/adgeo-24-125-2010.

Crawford, O. G. S., 1923. Air survey and archæology. *The Geographical Journal* 61(5), 342–360.

Curwen, E. and E. Curwen, 1923. Sussex lynchets and their associated fieldways. *Sussex Archaeological Collections* 64, 1–65.

Custer, J. F., T. Eveleigh, V. Klemas, and I. Wells, 1986. Application of landsat data and synoptic remote sensing to predictive

models for prehistoric archaeological sites: An example from the delaware coastal plain. *American Antiquity* 51(3), 572–588, doi: 10.2307/281753.

Davis, D., 2021. Theoretical repositioning of automated remote sensing archaeology: Shifting from features to ephemeral landscapes. *Journal of Computer Applications in Archaeology* 4, 94–109, doi: 10.5334/jcaa.72.

De Guio, A., L. Magnini, and C. Bettineschi, 2015. Geobia approaches to remote sensing of fossil landscapes: Two case studies from northern italy. *Across Space and Time, CAA Perth* .

De Laet, V., G. van Loon, A. Van der Perre, I. Deliever, and H. Willems, 2015. Integrated remote sensing investigations of ancient quarries and road systems in the Greater Dayr al-Barshā Region, Middle Egypt: A study of logistics. *Journal of Archaeological Science* 55, 286–300, doi:10.1016/j.jas.2014.10.009.

Deng, J., W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, 2009. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition* 248–255, doi:10.1109/CVPR.2009.5206848.

Dutta, A., A. Gupta, and A. Zissermann, 2016. VGG image annotator (VIA). http://www.robots.ox.ac.uk/ vgg/software/via/, version: 2.0.1, Accessed: 10-04-21.

Dutta, A. and A. Zisserman, 2019. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, New York, NY, USA: ACM, doi:10.1145/3343031.3350535, URL https://doi.org/10.1145/3343031.3350535.

Evans, D., 2016. Airborne laser scanning as a method for exploring long-term socio-ecological dynamics in cambodia. *Journal of Archaeological Science* 74, 164–175, doi:https://doi.org/10.1016/j.jas.2016.05.009.

Evans, D. H., R. J. Fletcher, C. Pottier, J.-B. Chevance, D. Soutif, B. S. Tan, S. Im, D. Ea, T. Tin, S. Kim, C. Cromarty, S. De Greef,

K. Hanus, P. Bâty, R. Kuszinger, I. Shimoda, and G. Boornazian, 2013. Uncovering archaeological landscapes at angkor using lidar. *Proceedings of the National Academy of Sciences* 110(31), 12595–12600, doi:10.1073/pnas.1306539110.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88(2), 303–338.

Garcia-Garcia, A., S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, 2018. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing* 70, 41–65, doi:https://doi.org/10.1016/j.asoc.2018.05.018.

Giffen, A. v., 1939. De zgn. heidensche legerplaats te zuidveld bij sellingen, gem. vlachtwedde. *Verslag Groninger Museum* 86–93.

Girshick, R., 2015. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–1448, doi:10.1109/ICCV.2015.169.

Girshick, R., J. Donahue, T. Darrell, and J. Malik, 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *arxiv* doi:10.1109/CVPR.2014.81.

Goodfellow, I., Y. Bengio, and A. Courville, 2016. *Deep Learning*. MIT Press, `http://www.deeplearningbook.org`.

Groenman-van Waateringe, W., 1992. Palynology and archaeology: the history of a plaggen soil from the veluwe, the netherlands. *Review of Palaeobotany and Palynology* 73(1), 87–98, doi:https://doi.org/10.1016/0034-6667(92)90047-K, festschrift For Professor Van Zeist.

Gulli, A., 2017. *Deep Learning with Keras*. Packt Publishing.

Guo, Y., 2017. Deep learning for visual understanding (dissertation).

Guo, Y., Y. Liu, T. Georgiou, and M. Lew, 2017. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval* 7, 87–93.

Guyot, A., L. Hubert-Moy, and T. Lorho, 2018. Detecting Neolithic burial mounds from LiDAR-derived elevation data using a multi-scale approach and machine learning techniques. *Remote Sensing* 10(2), 225–244, doi:10.3390/rs10020225.

Guyot, A., M. Lennon, T. Lorho, and L. Hubert-Moy, 2021. Combined detection and segmentation of archeological structures from lidar data using a deep learning approach. *Journal of Computer Applications in Archaeology* 4, 1, doi:10.5334/jcaa.64.

Hay, G. and G. Castilla, 2008. *Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline*, 75–89. doi:10.1007/978-3-540-77058-9_4.

He, K., G. Gkioxari, P. Dollar, and R. B. Girshick, 2017. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)* 2980–2988.

He, K., X. Zhang, S. Ren, and J. Sun, 2015. Deep residual learning for image recognition. *CoRR* .

Hein, L., 2011. Economic benefits generated by protected areas: the case of the hoge veluwe forest, the netherlands. *Ecology and society* 16(2), Art. 13–Art. 13.

Humme, A., R. Lindenbergh, and C. Sueur, 2006. Revealing Celtic Fields From LiDAR Data Using Kriging Based Filtering. *Proc. ISPRS Comm. V Symp.* 35(January), 25–27.

Juba, B. and H. S. Le, 2019. Precision-recall versus accuracy and the role of large data sets. *Proceedings of the AAAI Conference on Artificial Intelligence* 33(01), 4039–4048, doi:10.1609/aaai.v33i01.33014039.

Jung, A. B., K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte *et al.*, 2020. imgaug. `https://github.com/aleju/imgaug`, online; accessed 01-Feb-2020.

Kaymak, Ç. and A. Uçar, 2019. *A Brief Survey and an Application of Semantic Image Segmentation for Autonomous Driving*, 161–200. Cham: Springer International Publishing, doi:10.1007/978-3-030-11479-4_9.

Kazimi, B., F. Thiemann, and M. Sester, 2019. Object instance segmentation in digital terrain models. In M. Vento and G. Percannella (eds.), *Computer Analysis of Images and Patterns*, 488–495, Cham: Springer International Publishing.

Kingma, D. P. and J. Ba, 2014. Adam: A method for stochastic optimization. *CoRR* .

Kokalj, and M. Somrak, 2019. Why not a single image? combining visualizations to facilitate fieldwork and on-screen mapping. *Remote Sensing* 11(7), doi:10.3390/rs11070747.

Kooistra, M. and G. Maas, 2008. The widespread occurrence of celtic field systems in the central part of the netherlands. *Journal of archaeological science* 35(8), 2318–2328.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25.

Lambers, K., 2018. Airborne and Spaceborne Remote Sensing and Digital Image Analysis in Archaeology. In C. S. F. Bubenzer (ed.), *Digital Geoarchaeology: New Techniques for Interdisciplinary Human-Environmental Research*, 109–122, doi:10.1007/978-3-319-25316-9_7.

Lambers, K. and A. Traviglia, 2016. Automated detection in remote sensing archaeology: a reading list. *AARGnews* 53, 25–29.

Lambers, K., W. B. Verschoof-van der Vaart, and Q. P. J. Bourgeois, 2019. Integrating remote sensing, machine learning, and citizen science in dutch archaeological prospection. *Remote Sensing* 11(7), doi:10.3390/rs11070794.

Lasaponara, R. and N. Masini, 2018. Space-based identification of archaeological illegal excavations and a new automatic method for

looting feature extraction in desert areas. *Surveys in Geophysics* 39, doi:10.1007/s10712-018-9480-4.

Leisz, S., 2013. *An Overview of the Application of Remote Sensing to Archaeology During the Twentieth Century*, 11–19. doi:10.1007/978-1-4614-6074-9_2.

Lemmens, J., Z. Stančič, and R. Verwaal, 1993. Automated archaeological feature extraction from digital aerial photographs. In T. M. Andresen, J. and I. Scollar (eds.), *Computing the Past. Computer Applications and Quantitative Methods in Archaeology*, 45–52, Aarhus: Aarhus University Press.

Lillesand, T. and R. Kiefer, 2015. *Remote Sensing and Image Interpretation, 7th Edition*. International series of monographs on physics, Wiley, New York.

Lin, T., P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, 2017. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944, doi:10.1109/CVPR.2017.106.

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, 2014. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (eds.), *Computer Vision – ECCV 2014*, 740–755, Springer International Publishing.

Luo, L., X. Wang, H. Guo, R. Lasaponara, X. Zong, N. Masini, G. Wang, P. Shi, H. Khatteli, F. Chen, S. Tariq, J. Shao, N. Bachagha, R. Yang, and Y. Yao, 2019. Airborne and spaceborne remote sensing for archaeological and cultural heritage applications: A review of the century (1907–2017). *Remote Sensing of Environment* 232, doi:10.1016/j.rse.2019.111280.

Mantovan, L. and L. Nanni, 2020. The computerization of archaeology: Survey on artificial intelligence techniques. *SN Computer Science* 1, 267, doi:10.1007/s42979-020-00286-w.

Menze, B., J. Ur, and A. Sherratt, 2006. Detection of ancient settlement mounds: Archaeological survey based on the srtm ter-

rain model. *Photogrammetric Engineering Remote Sensing* 72, doi:10.14358/PERS.72.3.321.

Mikołajczyk, A. and M. Grochowski, 2018. Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 117–122, doi:10.1109/IIPHDW.2018.8388338.

Mohri, M., A. Rostamizadeh, and A. Talwalkar, 2012. *Foundations of Machine Learning*.

Nara, M., B. Mukesh, P. Padala, and B. Kinnal, 2019. Performance evaluation of deep learning frameworks on computer vision problems. 670–674, doi:10.1109/ICOEI.2019.8862603.

Nielsen, N. H. and K. Dalsgaard, 2017. Dynamics of celtic fields—a geoarchaeological investigation of Øster lem hede, western jutland, denmark. *Geoarchaeology* 32(3), 414–434, doi:https://doi.org/10.1002/gea.21615, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/gea.21615.

Opitz, R. and J. Herrmann, 2018. Recent Trends and Long-standing Problems in Archaeological Remote Sensing. *Journal of Computer Applications in Archaeology* 1(1), 19–41, doi:10.5334/jcaa.11, URL https://doi.org/10.5334/jcaa.11.

Orengo, H. A., F. C. Conesa, A. Garcia-Molsosa, A. Lobo, A. S. Green, M. Madella, and C. A. Petrie, 2020. Automated detection of archaeological mounds using machine-learning classification of multisensor and multitemporal satellite data. *Proceedings of the National Academy of Sciences* 117(31), 18240–18250, doi:10.1073/pnas.2005583117, URL https://www.pnas.org/content/117/31/18240.

O'Shea, K. and R. Nash, 2015. An introduction to convolutional neural networks. *ArXiv* abs/1511.08458.

Pan, S. J. and Q. Yang, 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10), 1345–1359, doi:10.1109/TKDE.2009.191.

Pang, B., E. Nijkamp, and Y. N. Wu, 2020. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics* 45(2), 227–248, doi:10.3102/1076998619872761, URL https://doi.org/10.3102/1076998619872761.

Perez, L. and J. Wang, 2017. The effectiveness of data augmentation in image classification using deep learning. *CoRR* abs/1712.04621.

Qian, N., 1999. On the momentum term in gradient descent learning algorithms. *Neural Networks* 12(1), 145–151, doi:https://doi.org/10.1016/S0893-6080(98)00116-6.

Redfern, S., 1998a. An approach to automated morphological-topographical classification. *AARGnews* 17, 31–37.

Redfern, S., 1998b. An approach to automated morphological-topographical classification. *AARGnews* 17, 31–37.

Ren, S., K. He, R. Girshick, and J. Sun, 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* .

Ren, S., K. He, R. Girshick, and J. Sun, 2017. Faster R-CNN. doi:10.1109/TPAMI.2016.2577031.

Risbøl, O. and L. Gustavsen, 2018. Lidar from drones employed for mapping archaeology – potential, benefits and challenges. *Archaeological Prospection* 25(4), 329–338, doi:https://doi.org/10.1002/arp.1712.

Robbins, H. and S. Monro, 1951. A stochastic approximation method. *The Annals of Mathematical Statistics* 22(3), 400–407.

Roccetti, M., L. Casini, G. Delnevo, V. Orrù, and N. Marchetti, 2020. Potential and limitations of designing a deep learning model for discovering new archaeological sites: A case with the mesopotamian floodplain. *Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good* 216–221, doi:10.1145/3411170.3411254.

Ronneberger, O., P. Fischer, and T. Brox, 2015. U-net: Convolutional networks for biomedical image segmentation. *CoRR* abs/1505.04597, URL http://arxiv.org/abs/1505.04597.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3), 211–252, doi:10.1007/s11263-015-0816-y.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural Networks* 61, 85–117, doi:https://doi.org/10.1016/j.neunet.2014.09.003.

Schneider, A., M. Takla, A. Nicolay, A. Raab, and T. Raab, 2015. A template-matching approach combining morphometric variables for automated mapping of charcoal kiln sites. *Archaeological Prospection* 22(1), 45–62, doi:https://doi.org/10.1002/arp.1497.

Shi, S., Q. Wang, P. Xu, and X. Chu, 2016. Benchmarking state-of-the-art deep learning software tools. *2016 7th International Conference on Cloud Computing and Big Data (CCBD)* 99–104, doi:10.1109/CCBD.2016.029.

Simonyan, K. and A. Zisserman, 2014. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556* .

Soroush, M., A. Mehrtash, E. Khazraee, and J. A. Ur, 2020. Deep learning in archaeological remote sensing: Automated qanat detection in the Kurdistan region of Iraq. *Remote Sensing* 12(3), 500–518, doi:10.3390/rs12030500.

Spek, T., W. G.-v. Waateringe, M. Kooistra, and L. Bakker, 2003. Formation and land-use history of celtic fields in north-west europe – an interdisciplinary case study at zeijen, the netherlands. *European journal of archaeology* 6(2), 141–173.

Szegedy, C., Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, 2015. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1–9, doi:10.1109/CVPR.2015.7298594.

Szeliski, R., 2011. *Computer vision algorithms and applications*. London; New York: Springer.

Traviglia, A. and D. Cottica, 2011. Remote sensing applications and archaeological research in the Northern Lagoon of Venice: The case of the lost settlement of Constanciacus. *Journal of Archaeological Science* 38(9), 2040–2050, doi:10.1016/j.jas.2010.10.024.

Traviglia, A. and A. Torsello, 2017. Landscape Pattern Detection in Archaeological Remote Sensing. *Geosciences* 7, 128, doi:10.3390/geosciences7040128.

Trier, D., S. Larsen, and R. Solberg, 2009. Automatic detection of circular structures in high-resolution satellite images of agricultural land. *Archaeological Prospection* 16(1), 1–15, doi:https://doi.org/10.1002/arp.339.

Trimble Germany, G., 2011. ecognition developer 8.7 reference book. document version 8.7.

Uijlings, J., K. Sande, T. Gevers, and A. Smeulders, 2013. Selective search for object recognition. *International Journal of Computer Vision* 104, 154–171, doi:10.1007/s11263-013-0620-5.

van der Heide, C. M., J. C. van den Bergh, E. C. van Ierland, and P. A. Nunes, 2008. Economic valuation of habitat defragmentation: A study of the Veluwe, the Netherlands. *Ecological Economics* 67(2), 205–216, doi:10.1016/j.ecolecon.2008.04.012.

van der Zon, N., 2013. Kwaliteitsdocument ahn2.

Verschoof-van der Vaart, W. B. and K. Lambers, 2019. Learning to Look at LiDAR: The Use of R-CNN in the Automated Detection of Archaeological Objects in LiDAR Data from the Netherlands. *Journal of Computer Applications in Archaeology* 2(1), 31–40, doi:10.5334/jcaa.32.

Verschoof-van der Vaart, W. B., K. Lambers, W. Kowalczyk, and Q. P. Bourgeois, 2020. Combining Deep Learning and Location-Based Ranking for Large-Scale Archaeological Prospection of LiDAR Data from The Netherlands. *ISPRS International Journal of Geo-Information* 9(5), 293, doi:10.3390/ijgi9050293.

Verschoof-van der Vaart, W. B. and J. Landauer, 2021. Using carcassonnet to automatically detect and trace hollow roads in lidar data

from the netherlands. *Journal of Cultural Heritage* 47, 143–154, doi: https://doi.org/10.1016/j.culher.2020.10.009.

Vletter, W. and R. Van Lanen, 2018. Finding vanished routes: Applying a multi-modelling approach on lost route and path networks in the veluwe region, the netherlands. *Rural Landscapes: Society, Environment, History* 5, 2, doi:10.16993/rl.35.

Voulodimos, A., N. Doulamis, A. Doulamis, and E. Protopapadakis, 2018. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience* 2018, 1–13, doi:10.1155/2018/7068349.

Whitefield, A., 2017. Neolithic 'celtic' fields? a reinterpretation of the chronological evidence from céide fields in north-western ireland. *European Journal of Archaeology* 20(2), 257–279, doi:10.1017/eaa.2016.5.

Yosinski, J., J. Clune, Y. Bengio, and H. Lipson, 2014. How transferable are features in deep neural networks? *CoRR* abs/1411.1792.

Yousefpour, R., M. Didion, J. B. Jacobsen, H. Meilby, G. M. Hengeveld, M.-J. Schelhaas, and B. J. Thorsen, 2015. Modelling of adaptation to climate change and decision-makers behaviours for the veluwe forest area in the netherlands. *Forest Policy and Economics* 54, 1–10, doi:https://doi.org/10.1016/j.forpol.2015.02.002.

Yu, Y., K. Zhang, L. Yang, and D. Zhang, 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on mask-rcnn. *Computers and Electronics in Agriculture* 163, 104846, doi:https://doi.org/10.1016/j.compag.2019.06.001.

Yuan, X., J. Shi, and L. Gu, 2021. A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Systems with Applications* 169, 114417, doi:https://doi.org/10.1016/j.eswa.2020.114417.

Zaccone, G. and M. Karim, 2018. *Deep Learning with TensorFlow - Second Edition.*

Zakšek, K., K. Oštir, and Kokalj, 2011. Sky-view factor as a relief visualization technique. *Remote Sensing* 3(2), 398–415, doi:10.3390/rs3020398.

List of Figures

List of Tables

Abstract

The flood of archaeological remote sensing data in present times calls for digital solutions which can reduce the time and cost required to manually analyse these vast amounts of data. In recent times, Deep Learning techniques based on Covolutional Neural Networks for automated detection of archaeological objects, are fast gaining traction due to the potential they hold. However, much of these studies remain restricted to detection of discrete objects with uniform morphologies. Thus, there lies a gap in the use of these methodologies for mapping of larger archaeological systems, which can contribute immensely to landscape archaeology, and our knowledge of human cultural activity in the past.

This thesis attempts to make this shift by using a CNN-based instance segmentation methodology to detect individual plots of large Celtic Field systems. It was implemented on LiDAR data from the Veluwe region in the Netherlands, using the Mask R-CNN model. The results show that the methodology has the ability to not only detect field plots present in the landscape, but also outline their exact shape. These results when embedded in a wider framework can contribute greatly to archaeological prospection and our understanding of the archaeological landscape in the Veluwe.

Appendix A

Glossary and Abbreviations used

LiDAR - Light Detection and Ranging: A remote sensing method which uses light in the form of a pulsed laser to measure ranges (variable distances) to the Earth
AI - Artificial Intelligence: a computer science discipline devoted training machines to perform tasks usually associated with human intelligence, without human intervention
ML - Machine Learning: a branch of AI, that can be defined as 'training' computational algorithms to detect patterns in past data, and subsequently make decisions and predictions regarding new data
DL - Deep Learning: a subset of Machine Learning, which utilises deep architecture
Computer Vision: a field of computer science which centers around the ability of computers to see and interpret digital images and videos.
Object detection: computer vision method which localises position of an object within the image through a bounding box and assigns a class label
Instance Segmentation: identifying the object in an image at the pixel level, and creating a binary mask which helps identify the exact shape of the object
CNN - Convolutional Neural Networks: the deep learning algorithms most commonly used for Computer Vision applications
Mask R-CNN - Mask Regional Convolutional Neural Networks: a state-of-the-art deep neural network developed for the purposes of solving instance segmentation problems

Appendix B

CNN Theory and Architectures

B.1 Convolutional Neural Networks

Deep Learning is a subset of Machine Learning, which utilises deep architecture of Artificial Neural Networks(ANNs). These are computational systems, built to mimic the learning process of the human brain. It consists of a number of processors called neurons, which are interconnected to one another and produce a sequence of real-valued activations, to collectively collect an input and optimise the output (Schmidhuber 2015, 86). A basic structure of ANN architecture can be seen in Figure B.1.

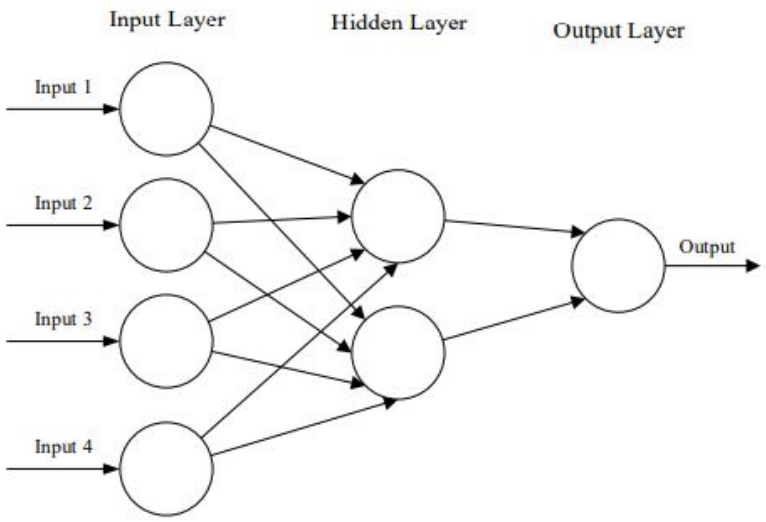


Figure B.1: Basic structure of an Artificial Neural Network(ANN) (O'Shea and Nash 2015, 2)

The input layer receives a combination of input values, in the form of information about the samples, and weights, which indicate the

importance of the inputs being fed into the neurons (refer to Figure B.2 for structure of a neuron) (Kaymak and Uçar 2019, 167). The transfer function then calculates the net input of the weights and input values (Kaymak and Uçar 2019, 167). Finally, the activation function processes the net input value calculated by the transfer function and generates an output, which in this case is distributed within the hidden layers of the ANN. The activation function is generally a non-linear function, and it transfers this non-linearity to the output as represented by the mathematical Equation B.1.

$$y = f(\sum_{i=1}^{n} W_i x_i + b) \tag{B.1}$$

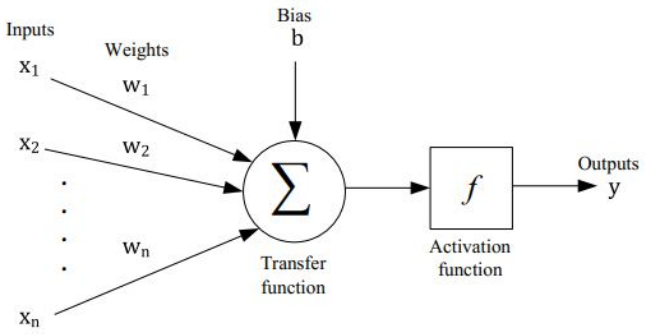


Figure B.2: Structure of an artificial neuron (Kaymak and Uçar 2019, 166)

The most commonly used activation function in deep learning and convolutional neural network algorithms is the Rectified Linear Unit or ReLU function. This activation function outputs the input value directly if it is positive, else it outputs a zero value (Kaymak and Uçar 2019, 168). It is represented by Equation B.2

$$f(x) = max(0, x) \tag{B.2}$$

Convolutional Neural Networks or CNNs are the deep learning algorithms most commonly used for Computer Vision applications. In simplistic terms, they can be described as a type of ANN system. There are, however, certain differences from traditional ANNs to take into account the main task of CNNs - pattern analysis and recognition within images. (O'Shea and Nash 2015). Traditional ANNs can-

not handle the computational complexity of computing image data. Through CNNs, we can reduce the model of parameters, and thus the complexity of the model, reducing a chance of overfitting (O'Shea and Nash 2015).

A CNN network is trained in two stages. The first stage, or the forward stage comprises of representing the input image with the associated weights for each layer(Guo 2017, 10). A loss cost is then calculated by comparing the predicted output with the ground truth labels (Guo 2017, 10). In the second stage, or the backward stage, the loss costs are used to re-update the parameters, which is followed by a another stage of forward transmission. This process continues for a defined number of iterations (Guo 2017, 10). When the entire dataset completes one cycle of forward and backward transmission, it is known as an epoch.

A CNN comprises of three main layers - convolutional layers, pooling layers and fully connected layers. The first two layers perform the task of feature learning and are generally placed in alternating layers. The last fully connected layers perform the task of assigning class labels to the input image (Voulodimos *et al.* 2018, 2). The basic structure of a CNN network can be seen in Figure B.3.

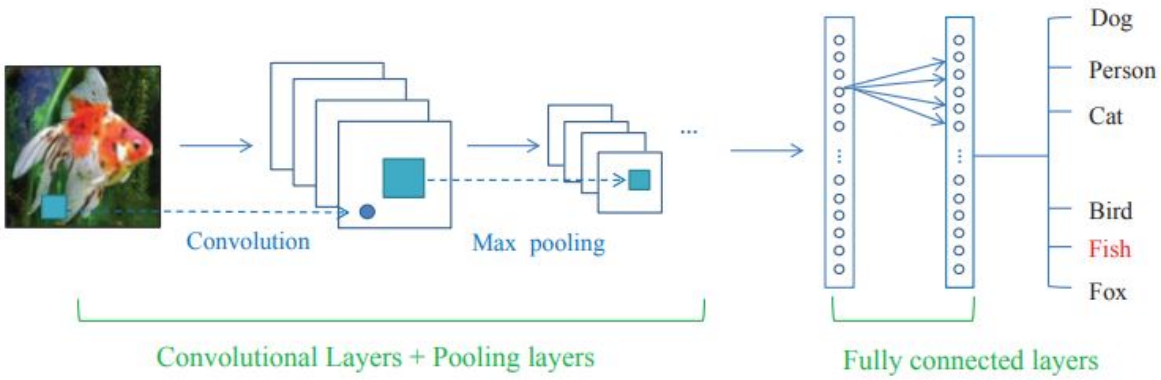


Figure B.3: Basic structure of a CNN architecture

1. Convolutional layer: The first layer comprises of extracting relevant features from the input image. This is done through a mathematical function called convolution(Voulodimos *et al.* 2018. It is used to learn image features by performing the function on small squares of input data. This function is defined as per Equation B.3(Goodfellow

et al. 2016, 327).

$$s(t) = (x * w)(t) = \int x(a)w(t-a)\,da \tag{B.3}$$

The first argument i.e. x is referred to an input, while the second one i.e. w is referred to as as filter matrix, which reduces the input image by extracting the necessary features. It is also known as a kernel or a feature map (Goodfellow *et al.* 2016; Voulodimos *et al.* 2018). As opposed to ANNs which have all fully connected layers, the CNN architecture substitutes the first few layers with a convolutional, in order to cut down learning time and accommodate the computational power of handling image inputs, by reducing parameters (Voulodimos *et al.* 2018, 3). The ReLU activation function is used to add non-linearity to the CNN, after the convolution function is applied.

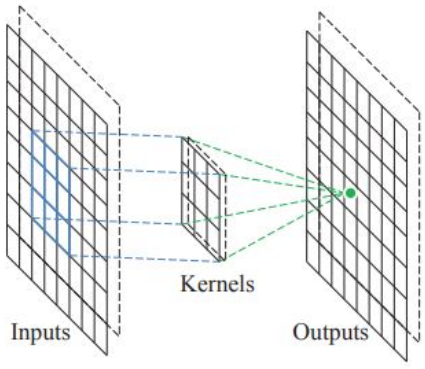


Figure B.4: Convolutional Layer (Guo 2017, 11)

2. Pooling Layer: These layers reduce spatial size of input volume i.e. width and height, by replacing output of a feature map with a net statistical summary (Goodfellow *et al.* 2016, 335). For example, the max pooling function performs a dimensional reduction by extracting the maximum value within a feature map (Goodfellow *et al.* 2016, 335). Other common types of pooling functions include average pooling and sum pooling. Along with reduction in size, there is also a loss in information in this process. However, this decrease leads to a reduction in computational power (Voulodimos *et al.* 2018, 3).

3. Fully Connected Layer: These layers contain 90% of the CNN parameters, and involve converting the 2D feature maps into a 1D

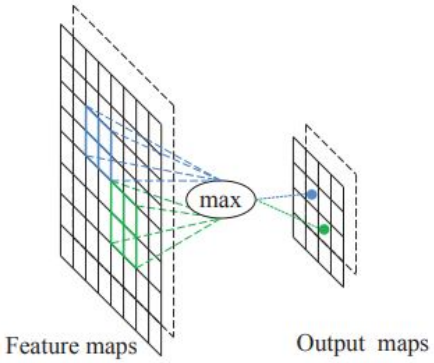


Figure B.5: Pooling Layer (Max pooling) (Guo 2017, 12)

feature vector(Guo 2017, 14). These vectors are then used for either a classification process or further processing.

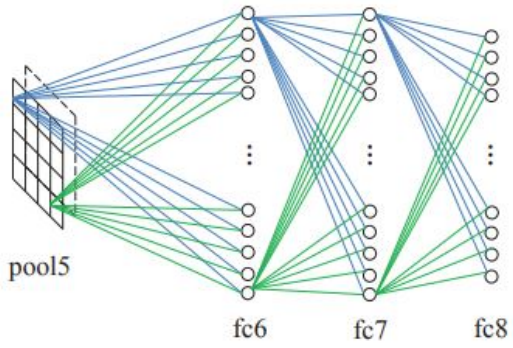


Figure B.6: Fully Connected Layer (Guo 2017, 14)

B.1.1 Object Detection

CNNs showed significant results with respect to image classification in 2012 (Krizhevsky *et al.* 2012), on the ImageNet Large Scale Visual Recognition Challenge (Deng *et al.* 2009; Russakovsky *et al.* 2015). However, object detection differs from classification as it also requires localisation of objects within an image. To solve this issue, R-CNN or Regions with CNN features was developed (Girshick *et al.* 2014). In this method, a number of category-independant region proposals are generated through a search algorithm known as selective search(Uijlings *et al.* 2013). Consequently, A CNN network extracts relevant features from each region proposal, and these are then fed

into a Support Vector Machine(SVM) for classification (Girshick *et al.* 2014). This method however is computationally expensive. Therefore Fast R-CNN was developed to reduce the computational time, and also improve detection. In Fast RCNN, as opposed to passing each generated interest through the CNN, the whole image is passed as input to produce a convolutinal feature map, thus sharing the computation. A Region of Interest (ROI) pooling layers extracts feature vectors from the feature map, based on the object proposals (Girshick 2015).

Faster R-CNN further brings down computation time by replacing the Selective Search algorithm CNN with a Region Proposal Network (RPN). In this case, the feature maps generated by the CNN are fed as inputs to the RPN, to generate proposals, thus making the process relatively cost free. RPNs can predict region proposals of multiple aspect ratios and scales by using anchor boxes as reference. The region proposals are then passed through ROI pooling and then classified (Ren *et al.* 2017).

The Faster RCNN architecture is used for the object detection part of the Mask RCNN algorithm used in this thesis.

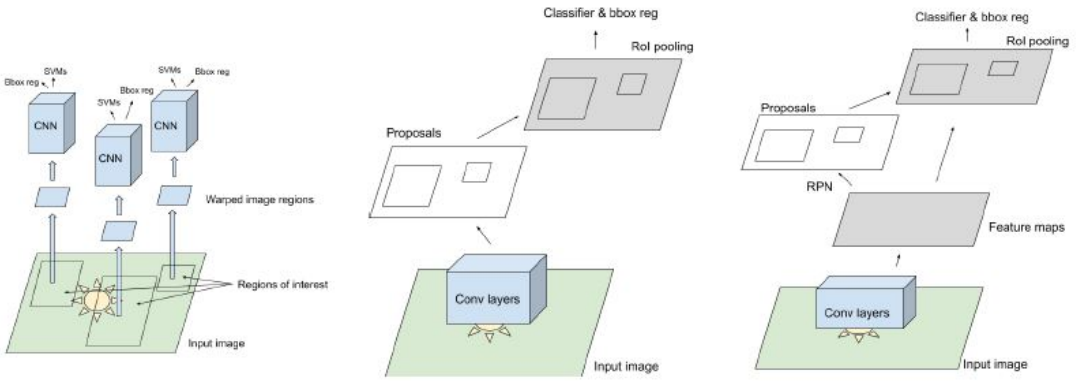


Figure B.7: Architectures of R-CNN, Fast R-CNN & Faster R-CNN

B.1.2 Semantic Segmentation

Image segmentation has been one of the more complex and deeper forms of computer vision. This is due to the fact that in this approach, it is not always necessary to know the exact identity of the

objects being detected. A semantic segmentation algorithm clusters pixels within an image that belong together semantically, and embeds the spatial information of objects. Therefore, each pixel within the image is assigned a class label, rather than detecting complete objects. These algorithms can be roughly divided into three categories - region-based, FCN-based and weakly supervised (Guo *et al.* 2017). The most successful technique out of these is the FCN or Fully Connected Network technique. Another common deep learning semantic segmentation architectures is UNet (Ronneberger *et al.* 2015).

Image segmentation provides some benefits over object detection. It improves efficiency, as only specific segments of the image are taken into account. It also enhances accuracy, as it eliminates problems relation to background noise, which is the case with window based object detection (Guo *et al.* 2017, 90).However there are also some limitations to the method. It does not take into account the general overall context in its pixel-wise, approach. Moreover, there is no instance-awareness of different objects of the same type (Garcia-Garcia *et al.* 2018, 9). In case of remote sensing imagery specifically, high levels of pixel accuracy is required, as almost every object contains meaningful information, thus causing problems in delineation of object boundaries (Yuan *et al.* 2021). Semantic segmentation has in the past been used for various remote sensing applications, such as environmental monitoring, crop analysis and land use in urban spaces.

B.1.3 Instance Segmentation

Instance segmentation is a method which tackles both object detection and semantic segmentation. It involves the prediction of object instances, AND producing their pixel-wise segmentation mask. It differs from semantic segmentation in that it delineates each individual object instance within a category. An example can be seen in Figure B.8.

Mask RCNN - It adopts the detect-then-segment approach, first perform object detection to extract bounding boxes around each object instances, and then perform binary segmentation inside each

(a) Input Image (b) Semantic Segmentation (c) Instance Segmentation

Figure B.8: Types of Image Segmentation (https://analyticsindiamag.com)

B.2 Mask R-CNN

Mask R-CNN (or Mask Regional Convolutional Neural Network) is a state-of-the-art deep neural network developed for the purposes of solving instance segmentation problems. It was first proposed in 2018, as a framework which extends the Faster R-CNN object detection neural network, by adding an extra branch which predicts a binary mask for the object being detected, in addition to a bounding box and class labels (He *et al.* 2017). Thus the segmentation process occurs parallel to classification and detection. The framework consists of mainly three stages - extracting feature maps, generating ROIs through a *Region Proposal Network* and finally using the generated ROIs to perform instance segmentation and object detection, through a fully convolution network (FCN). The difference from Faster R-CNN lies in the use of Feature Pyramis Networks (FPNs), replacing the *ROIPool* layer with *ROIAlign*, and the introduction of the mask branch.

B.2.1 Backbone: ResNet+FPN

The 'backbone architecture' of a deep neural network refers to the starting convolutional layers used for initial feature extraction from the input image. Previous challenges and research has shown that the depth of backbone network architecture is crucial for model performance (Simonyan and Zisserman 2014; Szegedy *et al.* 2015). However, it was discovered that adding more layers to the network also led to a degradation in model performance (He *et al.* 2015, 1). This problem was mitigated with the advent of *residual blocks*, used to create *Residual Networks* or *ResNet*. An image of a residual block can be seen

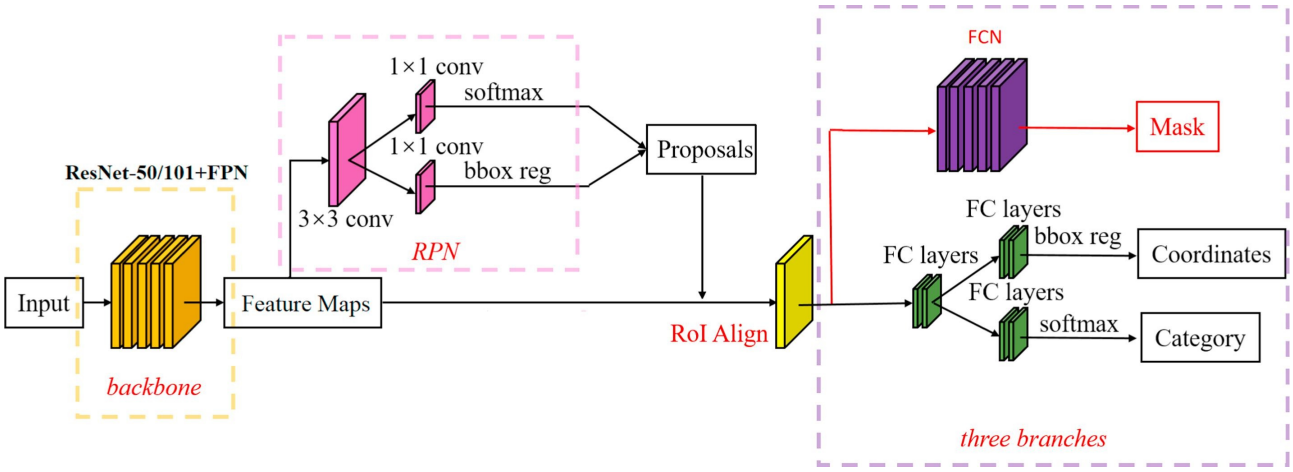


Figure B.9: Mask R-CNN architecture, with ResNet+FPN backbone (Yu *et al.* 2019, 4) (Guo 2017, 11)

in Figure B.10.

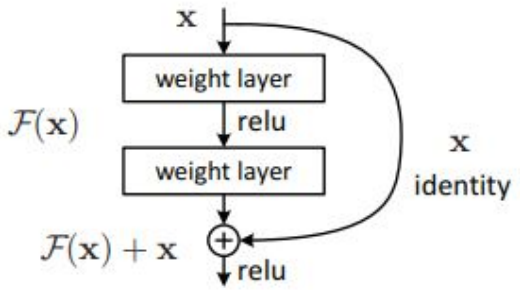


Figure B.10: Structure of a residual block) (He *et al.* 2015, 2)

The theory applied is that the additional layers added to make a shallow network "deep" are simply *identity mapping* i.e. returning the same arguments as the other 'original' layers. Thus a "shortcut-connection" is applied, wherein the additional layers are skipped, their identity mapping is simply done through an applied function and the outputs added to the original, stacked layer outputs (He *et al.* 2015, 2). Networks with these residual blocks were seen to have improved model accuracy over those without (He *et al.* 2015). Mask R-CNN uses either *ResNet-50* or *ResNet-101* (50 and 101 layers respectively) backbone architecture (He *et al.* 2017).

In addition to ResNet, another backbone architecture called a *Fea-*

ture Pyramid Network (FPN) was also used. One of the main challenges in computer vision is accounting for the difference in scales in object instances(Adelson *et al.* 1983). Featurized image pyramids were conceptualised to solve this problem. A pyramid of features is constructed, and a change in object scale is accounted for by shifting its level in the pyramid. In a normal featurized pyramid, the feature maps in the initial levels of the pyramid are made up of low-level structures. Therefore an FPN architecture has been used for R-CNN, which applies higl-level semantics throughout the pyramid. This done by constructing these pyramids with a bottom-up and top-down pathway. The former constitutes the general forward computation of the backbone CNN, whilst the latter constructs a higher resolution layers. There are also lateral connections in between, which improve detection and efficiency by acting as shortcut connections (similar to the residual blocks in ResNet) (Lin *et al.* 2017).

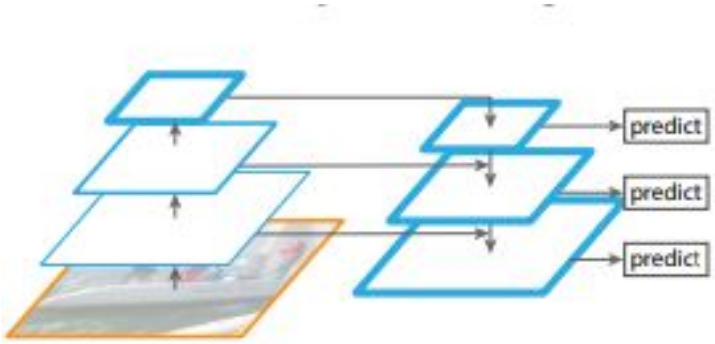


Figure B.11: Feature Pyramid Network) (Lin *et al.* 2017); the arrows represent the bottom-up, top-down and lateral connections

B.2.2 Stage I: Regional Proposal Network (RPN)

The RPN stage of the architecture deals with the actual "detection" of object instances. It outputs a set of rectangular object proposals (regressor layer), along with a score evaluating object detection capability with reference to the background class, known as objectness score (classifier layer) (Ren *et al.* 2016, 3).

The generation of region proposals is done by using a 'sliding window' over the feature map of the last shared convolutional layer. At each location in a sliding window, multiple region proposal are gen-

erated. An anchor forms the central point of the window, and has an association to an aspect ratio (width of image/height of image) and scale (size of image) (Ren *et al.* 2016, 4). The novelty of this system is that it is *translation-invariant* & *scale-invariant*, thus reducing model size and cost. The former means that the network can accommodate for a translation in the object within the image (for example, if it is rotated or flipped upside down and so on). The later account for multiple scales and aspect ratios of anchor boxes, by creating a *pyramid of anchors* (Ren *et al.* 2016, 4).

In Fast & Faster R-CNN, the next stage of extracting a features from a region of interest, and converting it into a fixed-length feature vector was done using the ROIPool operation (Girshick 2015; Ren *et al.* 2016). This method worked well for the object detection applications of these networks, however they were not aligned for pixel-to-pixel alignment. As a result, it performs a more coarse spatial quantization for feature extraction, causing misalignment between the ROI & (extracted) features. This in turn has an adverse effect on attaining pixel-level accuracy (He *et al.* 2017, [1,3]. *ROIAlign* in turn fixes this misalignment, by having a less harsh quantization proces & preserves exact spatial locations(He *et al.* 2017, 3). Replacing the ROIPool layer with ROIAlign showed upto a 50% increase in mask accuracy (He *et al.* 2017, 2).

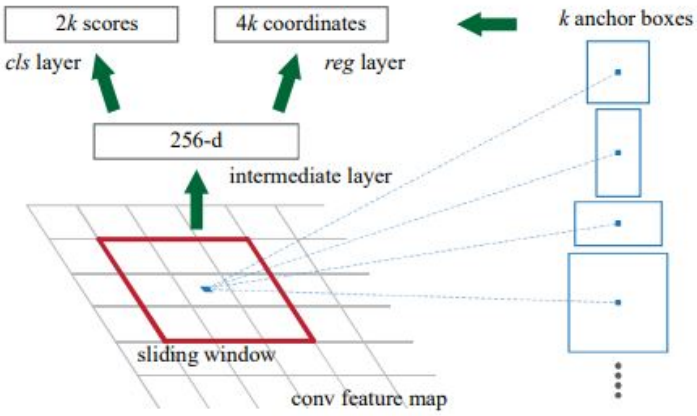


Figure B.12: Regional Proposal Network) (Ren *et al.* 2016)

B.2.3 Stage II: Network heads

Box Head: The output from the ROIAlign stage is reshaped, and then passed through a predictor stage with two branches, each comprising of a fully connected layer. The first branch is the classifier branch, which outputs the class label, and the second is the regressor layer, which outputs bounding box co-ordinates.

Mask Head: Parallel to the box head, the outputs from the ROI-Align stage are also sent to the mask branch. An (mxm) mask is predicted from each ROI through a fully connected network (FCN). This process preserves the spatial layout of the image object (He *et al.* 2017, 3).

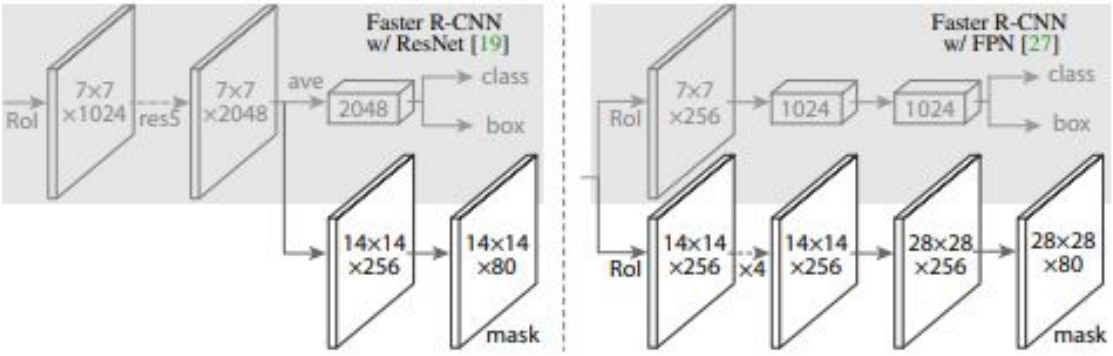


Figure B.13: Head Architecture for Mask R-CNN (ResNet & FPN) (He *et al.* 2017, 4)

Appendix C

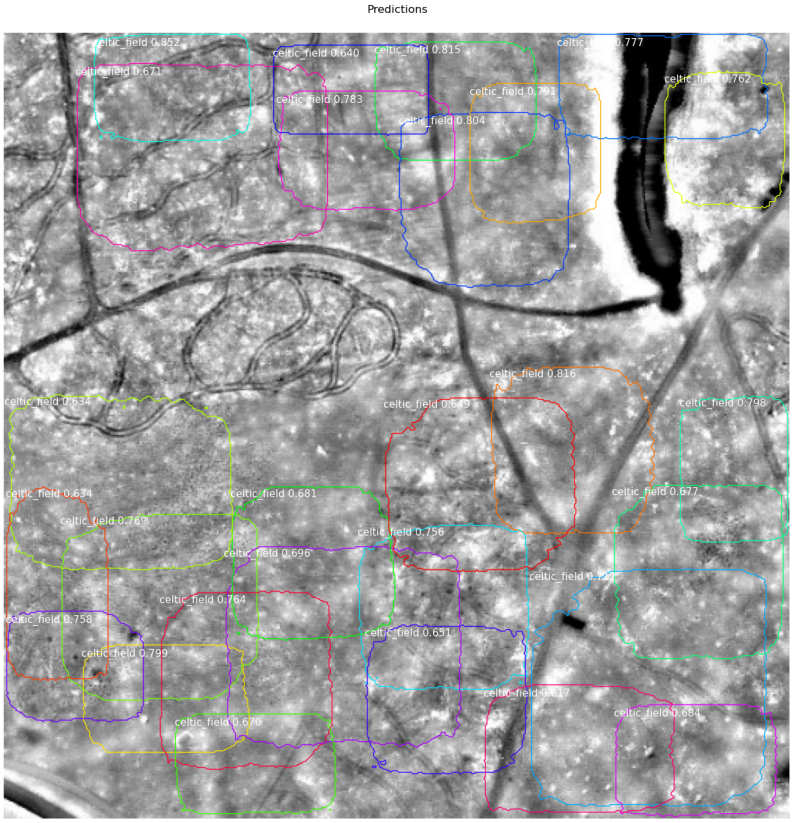
Default configurations of Matterport's Mask R-CNN

GPU_COUNT = 1 IMAGES_PER_GPU = 2 STEPS_PER_EPOCH
= 1000 # Number of training steps per epoch
VALIDATION_STEPS = 50 BACKBONE = "resnet101" BACKBONE_STRIDES
= [4, 8, 16, 32, 64] FPN_CLASSIF_FC_LAYERS_SIZE = 1024 TOP$_DOWN_PYRAM$
256
$NUM\_CLASSES = 1$
$RPN\_ANCHOR\_SCALES = (32, 64, 128, 256, 512)$
$RPN\_ANCHOR\_RATIOS = [0.5, 1, 2]$
$RPN\_ANCHOR\_STRIDE = 1$
$RPN\_TRAIN\_ANCHORS\_PER\_IMAGE = 256$
$PRE\_NMS\_LIMIT = 6000$
$POST\_NMS\_ROIS\_TRAINING = 2000$
$POST\_NMS\_ROIS\_INFERENCE = 1000$
$USE\_MINI\_MASK = True$
$IMAGE\_RESIZE\_MODE = "square"$
$IMAGE\_MIN\_DIM = 800$
$IMAGE\_MAX\_DIM = 1024$
$IMAGE\_CHANNE\_COUNT = 3$
$MEAN\_PIXEL = np.array([123.7, 116.8, 103.9])$
$TRAIN\_ROIS\_PER\_IMAGE = 200$
$ROI\_POSITIVE\_RATIO = 0.33$
$POOL\_SIZE = 7$
$MASK\_POO\_SIZE = 14$
$MASK\_SHAPE = [28, 28]$

$MAX\_GT\_INSTANCES = 100$
$RPN\_BBOX\_STD\_DEV = np.array([0.1, 0.1, 0.2, 0.2])$
$BBOX\_STD\_DEV = np.array([0.1, 0.1, 0.2, 0.2])$
$DETECTION\_MAX\_INSTANCES = 100$
$DETECTION\_MIN\_CONFIDENCE = 0.7$
$DETECTION\_NMS\_THRESHOLD = 0.3$
$LEARNING\_RATE = 0.001$
$LEARNING\_MOMENTUM = 0.9$
$WEIGHT\_DECAY = 0.0001$
$LOSS\_WEIGHTS =$
"rpn_class_loss" : 1.,
"rpn_bbox_loss" : 1.,
"$mrcnn\_class\_loss$" : 1.,
"$mrcnn\_bbox\_loss$" : 1.,
"$mrcnn\_mask\_loss$" : 1.

Appendix D

Prediction using Object Detection Annotations



We see that the masks created do not properly outline the shape

of the Celtic Fields. This probably because in the PASCAL VOC format used for object detection, the annotations are in the form of bounding boxes that cover the entirety of the object. Therefore, during the training process the algorithm probably 'learned' the extra areas within the bounding boxes but outside of the Celtic Field, and thus detected accordinglu during inference.