

Faculté des bioingénieurs

To what extent are field data dispensable for training the ResUNet-a neural network in automatic field boundaries delineation ?

Autrice :	Zoé Kempenaers
Promoteur :	Pierre Defourny
Lecteurs :	Julien Radoux Xavier Draye
Année Académique :	2023 - 2024
Bioingénieur :	Sciences et technologies de l'environnement

Acknowledgements

I am sincerely grateful to Professor Pierre Defourny for providing me with the opportunity to delve into this fascinating and challenging subject. His knowledge has been instrumental in allowing me to explore the world of remote sensing during my university studies.

A special thanks to Quentin Deffense for introducing me to deep learning and for his valuable advice throughout this work. His expertise and availability were crucial in shaping the direction and quality of this study.

I would also like to express my appreciation to the researchers and workers of the Geomatic research lab of UCLouvain. Their willingness to answer my questions and assist with various computer problems was immensely helpful.

Heartfelt thanks to my friends, who have made my university studies more enjoyable and continue to bring light, fun, and meaning to my daily life.

Lastly but not least, I want to extend my gratitude to my family for their unwavering support, especially my mother, who has consistently believed in me throughout my studies despite its non-linearity. Thank you for allowing me the time and space to find my way, change directions, and face doubts without question. My path is now clearer, and I'm moving forward with certainty.

Contents

1	Introduction	1
2	Literature review	3
2.1	Image segmentation	3
2.1.1	Edge-based vs. region-based methods	4
2.1.2	Canny operator	5
2.1.3	Watershed segmentation	5
2.1.4	Multi-resolution segmentation	6
2.2	Instance and semantic segmentations	7
2.2.1	Pixel-based vs. object-based classifications	7
2.2.2	Classifiers	8
	Random Forests	8
	Support Vector Machines	9
	Convolutional Neural Networks	10
2.3	Convolutional neural networks	10
2.3.1	How a neural network works	11
2.3.2	Key-concepts of neural networks	12
	Activation function	12
	Loss function	13
	Gradient descent	13
	Back propagation	13
2.3.3	Typical architecture of a convolutional neural network	14
	Convolutional layer	14
	Pooling layer	16
2.3.4	Description of ResUNet-a architecture	16
	A U-Net backbone architecture	16
	ResUNet-a architecture	18
	Loss function	20
2.4	Image segmentation evaluation metrics	21
2.4.1	Pixel-based metrics	22
	Precision and Recall	22

	Overall Accuracy	22
	F1-Score	23
	Matthews Correlation Coefficient	23
2.4.2	Object-based metrics	23
	Intersection over Union	23
	RecRate _{soft}	24
2.5	Related work	26
3	Objectives	29
4	Material and methods	31
4.1	Study area	31
4.1.1	Brabant and Hennuyer plateaus	31
4.1.2	Hesbaye	32
4.1.3	Ardennes	32
4.2	Study period	33
4.3	Data collection	33
4.3.1	Satellite data	33
4.3.2	LPIS	34
4.4	Preprocessing	34
4.4.1	Cloud mask application	35
4.4.2	Normalised Difference Vegetation Index calculation	35
4.4.3	Composites	36
4.5	eCognition segmentation	38
4.5.1	Software	38
4.5.2	Field segmentation and classification	38
4.5.3	Parameters	39
4.6	ResUNet-a	42
4.6.1	Method overview	42
4.6.2	Model	45
4.6.3	Instance mask and training data creation	45
4.6.4	Training	46
4.6.5	Inference	47
4.6.6	Instance segmentation for individual fields extraction	48
4.6.7	Performance evaluation	49
5	Results	51
5.1	Training	51
5.2	Performance evaluation	53

5.2.1	Scratch projects	56
5.2.2	Mix projects	57
5.2.3	Pre-training	57
5.2.4	Training from scratch vs. transfer learning	58
5.2.5	eCognition vs. ResUNet-a	60
6	Discussion	63
6.1	Effect of training dataset reduction	63
6.2	Effectiveness of pre-training	64
6.3	Effectiveness of deep learning	64
6.4	Limitations and error sources	65
6.5	Perspectives	66
7	Conclusion	67
	Bibliography	69

List of Figures

2.1	Example of fully connected neural network	12
2.2	Sigmoid and ReLU : two of the most commonly employed activation functions	12
2.3	Standard CNN architecture	14
2.4	Standard representation of a convolution matrix with stride and padding hyperparameters	15
2.5	Example of max pooling	16
2.6	U-Net architecture (Ronneberger et al., 2015)	17
2.7	The ResUNet-a architecture used in the context of field delineation . . .	19
2.8	Confusion matrix	21
2.9	Illustration of the predicted bounding box B_p and the ground-truth bounding box B_a associated with the IoU score (Amidi and Amidi, 2019)	24
4.1	Location of the three training and inference zones forming the study area in Wallonia	32
4.2	Overview of the preprocessing workflow	35
4.3	The 5-layer NDVI composites of the three training zones and their cut-outs	37
4.4	Overview of the eCognition segmentation	38
4.5	Illustration of the segmentation and classification stages performed on eCognition software	41
4.6	Overview of the method used from the training to the performance evaluation of ResUNet-a	43
4.7	Summary of the inputs of each project	44
4.8	Example of a 256x256 chip with its related extent, boundary and distance masks	46
4.9	Illustration and location of the 5-layer NDVI composite used for inference	47
5.1	Evolution of the different metrics during the 110 epochs of training where the red line indicates the optimal model according to the MCC Extent . .	52

5.2	Example of results for the entire inference zone. From left to right: the NDVI monthly composite of April 2019, the prediction obtained for the Mix 100 project and the associated instance segmentation compared with LPIS 2019	53
5.3	Zoom on the NDVI monthly composite of April 2019 and the predictions obtained for Mix and Scratch projects trained with 2% and 100% of field data, as well as the associated instance segmentations compared with LPIS 2019	55
5.4	Evolution of pixel-based and object-based performance metrics according to the number of labelled fields used for training	59
5.5	Illustration of the eCognition segmentation performed on the inference zone with the same method and parameters as the training zones	60

List of Tables

4.1	Spectral bands of Sentinel-2 and its resolutions	34
4.2	Parameters used for the MRS at the pixel and object levels	40
5.1	Number of field plots per zone used for training according to the project index, i.e. the percentage of field data used for training	52
5.2	Performance assessment on the pixel- and object-based metrics for the projects from scratch, i.e. without pre-training	56
5.3	Performance assessment on the pixel- and object-based metrics for the Mix projects, i.e. with pre-training	57
5.4	Performance assessment on the pixel- and object-based metrics for the pre-training	57
5.5	Comparison between eCognition, Pre-training, Scratch 2, and Mix 2 predictions	60

List of Abbreviations

BN	Batch Normalisation
CAP	Common Agricultural Policy
CISM	Center for High Performance Computing and Mass Storage
CNN	Convolutional Neural Network
GEOBIA	Geographic Object-Based Image Analysis
HSV	Hue-Saturation-Value
IoU	Intersection over Union
KNN	K-Nearest-Neighbors
LPIS	Land Parcel Identification System
MCC	Matthews Correlation Coefficient
MRS	Multi-resolution Segmentation
NDVI	Normalised Difference Vegetation Index
NIR	Near-Infrared
NN	Neural Network
OA	Overall Accuracy
OBIA	Object-Based Image Analysis
PSP	Pyramid Scene Parsing
ReLU	Rectified Linear Unit
RF	Random Forest
RGB	Red Green Blue
SCL	Scene Classification
SVM	Support Vector Machine
VHR	Very High Resolution
WS	Watershed Segmentation

Chapter 1

Introduction

Agriculture serves as the cornerstone of our society's food supply, naturally establishing itself as a pivotal economic sector. However, while the sector is of vital importance on a global scale, developing countries face unique challenges that are often an obstacle to their agricultural progress. Limited access to modern technology, insufficient financial resources, and inadequate infrastructure create barriers to the implementation of effective agricultural policies.

Accurate and up-to-date agricultural monitoring systems have become indispensable for the effective implementation of agricultural policy. A critical element of these systems lies in precisely determining the spatial extent and location of agricultural fields. Mapping the agricultural landscape serves as a valuable tool for predicting yields and undertaking long-term planning, ultimately enhancing the resilience of agriculture. Gathering this data facilitates the calculation of crop yields, identification of crop types, the creation of seasonal crop maps, compilation of agricultural statistics and development of host of other applications. Using the example of the Common Agricultural Policy (CAP) in European countries, it can also play a vital role in monitoring farmers' practices and assessing their eligibility for subsidies and support.

At present, the census of agricultural parcels in Europe is carried out by farmers themselves, who delineate the boundaries of their parcels and provide information regarding their contents. This census is aggregated in a collective database commonly known as the Land Parcel Identification System (LPIS). Each country in the European Union has its own LPIS, which is updated annually. However, the issue with this traditional method lies in its cost, time-consuming nature, and susceptibility to human errors. Integrating such a method into developing countries therefore seems an impractical task.

Meanwhile, there has been an ongoing effort to automate repetitive tasks in the field of computing. In recent years, machine learning has emerged as a central player in this endeavor, introducing a variety of new algorithms and innovative approaches.

Satellite remote sensing researchers have naturally risen to the occasion to address the challenges inherent in their discipline. The semantic labeling of satellite images holds significant promise across a spectrum of applications, including land-use planning, land-cover analysis, deforestation, and numerous others. Such applications involve laborious repetitive tasks that, if automated, could see the field of remote sensing advancing at an unprecedented rate. One compelling illustration of the effectiveness of machine learning in the context of remote sensing is the automated detection of field boundaries. In fact, considering an alternative data collection method for agricultural plots could enhance the efficiency of agriculture monitoring systems, especially for developing countries.

Automated extraction of field boundaries can not only strengthen agricultural monitoring systems in developed countries, but also and above all provide developing countries with access to essential decision-support tools, making a significant contribution to improving food security. In regions where resources are limited, but where it is essential to develop agricultural policies that guarantee the resilience of production systems, this innovative approach appears to be a fundamental and transformative solution. By taking advantage of automated technology, developing countries can overcome their limitations and make informed decisions that promote sustainable agricultural practices and, consequently, strengthen global food security.

Several studies have been carried out on the automatic detection of agricultural parcels. The use of convolutional neural networks has shown state-of-the-art results in this area, but in order to be effective, these artificial intelligence models must first undergo training, which requires a very significant amount of input field data. The objective of this master's thesis is to assess the extent to which field data required for deep learning training can be minimised by substituting it with data generated through a conventional, unsupervised image segmentation algorithm accessible on eCognition software. This evaluation is performed using the ResUNet-a convolutional neural network. The chosen study area is situated in Wallonia, Belgium, leveraging the accessibility of data derived from its LPIS, known as the anonymous agricultural parcel system.

Chapter 2

Literature review

Computer vision is a captivating and rapidly evolving field at the intersection of artificial intelligence (AI) and image processing. At its core, computer vision strives to replicate human vision, enabling computers to recognize objects, detect patterns, and make sense of visual data. Through the use of sophisticated algorithms and machine learning techniques, computer vision systems can identify objects (Deng et al., 2020), track movements (Krishna et al., 2021), gauge distances (Aldalil, 2021), and even assess the facial expression of individuals depicted in images (Canedo and Neves, 2019).

Machine learning, an integral part of computer vision, plays a crucial role in enabling computers to *learn* from data. This means that instead of being explicitly programmed to perform tasks, computer vision systems can be trained on vast datasets, via the development of a *model*, to recognise patterns and make predictions (Theodoridis, 2020).

2.1 Image segmentation

In general terms, image segmentation involves the task of separating objects within an image. Before the advent of machine learning, various more traditional methods were used for this purpose. These algorithms aimed to divide images into meaningful regions or objects based on specific criteria, such as color, intensity, or texture (Cremers et al., 2007; Muñoz et al., 2003). While these traditional approaches represented important milestones in computer vision, they often faced limitations in handling object occlusions, perspective and obscured boundaries with respect to other image parts in real-world scenes (Wang et al., 2022b).

As the demand for more precise and detailed image analysis grew across diverse fields, the computer vision community recognised the need for more advanced segmentation techniques. This need gave rise to instance and semantic segmentations, which led to a new era of image understanding. These techniques surpassed the capabilities of their predecessors by not only identifying objects and regions but also differentiating

between individual instances of objects and providing pixel-level labeling of images.

These traditional techniques are still used for a variety of reasons. The first is that, for Earth observation scientists wishing to use more advanced models such as those enabled by deep learning, the barriers to entry remain high (Hoeser and Kuenzer, 2020), but also because various softwares used in remote sensing provide easy access to these classic segmentation methods. Another reason is that these methods have lower computational costs and generally do not require training data, which is considered the biggest limitation of machine learning. The last reason is that these traditional methods are still of interest, and come into play at various stages even when machine learning is used, in order to boost the performances (Wang et al., 2022b). For example, the use of such methods can occur at the post-processing stage, as was the case in the Waldner and Diakogiannis (2020) study for the extraction of field boundaries. While the ResUNet-a convolutional neural network is capable of strictly identifying field boundaries, it is not necessarily capable of generating closed boundaries. Therefore, two post-processing methods are proposed in their study to extract individual fields: the first is the threshold-based cutoff method and the second is watershed segmentation.

2.1.1 Edge-based vs. region-based methods

Segmentation techniques rely on two fundamental pixel properties concerning their local neighborhood: discontinuity and similarity. The presence of pixel discontinuity leads to the development of edge-based methods, while the existence of pixel similarity gives rise to region-based methods. In practical applications, a hybrid approach combining both edge and region informations is often used to improve segmentation accuracy (Muñoz et al., 2003).

Edge-based segmentation focuses on identifying abrupt changes in intensity, color or texture within an image. These changes indicate potential boundaries between different objects or regions. Common techniques involve the use of edge detection operators such as Sobel, Canny or Scharr. The output of edge-based segmentation is often a binary edge map that highlights detected boundaries at the pixel level (Kaganami and Beiji, 2009). While effective at highlighting discontinuities, edge-based methods can be sensitive to noise and struggle to handle occlusions because they focus primarily on detecting boundaries without explicitly defining regions (Cremers et al., 2007).

Region-based segmentation, on the other hand, focuses on grouping pixels into regions with similar characteristics. This can include similarities in intensity, color or

texture. Methods such as thresholding, clustering and region growing algorithms are used to identify and group pixels that share these common properties (Kaganami and Beiji, 2009; Zhu and Yuille, 1996; Mittal et al., 2022). Region-based segmentation tends to be more robust to noise because it considers the overall properties of a group of pixels rather than individual pixels (Zand et al., 2016).

2.1.2 Canny operator

Several edge-based methods are available, ranging from basic algorithms such as Sobel or Scharr to more complex techniques such as Canny (Shrivakshan and Chandrasekar, 2012). The Canny edge operator is considered as the most performative edge detector compared to others, and is frequently employed as a standard for evaluating alternative contour-based operators (Watkins and van Niekerk, 2019).

The Canny method employs a multi-step procedure that can be summarised as follows : image filtering, image gradient graduation, non-maximum suppression, checking and connecting edges (Rong et al., 2014). The Canny edge detector starts the process by smoothing the image to remove any noise. It then calculates the gradient of the image to emphasise areas with high spatial derivatives. The algorithm then traverses along these regions, suppressing pixels that do not reach the maximum value (non-maximum suppression). The gradient array is now further reduced through hysteresis. Hysteresis is used to track along the remaining pixels that haven't been suppressed. This involves the use of thresholds: if the value exceeds a certain threshold, it's considered to be an edge (Juneja and Sandhu, 2009).

2.1.3 Watershed segmentation

Traditionally, watershed segmentation (WS) is primarily applied to elevation maps but can also be extended to non-elevation data (Beucher and Meyer, 1993). The watershed transform is a region-based segmentation approach that operates on a grayscale image with a single layer. The underlying concept draws inspiration from geography, envisioning a landscape or topographic relief being submerged by water, where watersheds represent the divide lines between regions influenced by rainfall (Young, 1983 ; Roerdink and Meijster, 2003).

In practice, there isn't a single algorithm claiming to perform watershed segmentation. Over time, various alternative algorithms have been developed to address specific needs and enhance performance in different applications. Notably, these developments aim to mitigate challenges like over-segmentation and interference, which are

issues encountered with the traditional watershed segmentation algorithm (Wu and Li, 2022).

As a result, multiple algorithms assert the generation of watersheds, but they often produce distinct objects. Watershed approaches can be topographic, viscous, stochastic, with or without apparent contours, and can be defined on images where nodes are weighted or on graphs where edges are weighted (Meyer, 2012).

2.1.4 Multi-resolution segmentation

The multi-resolution segmentation (MRS) algorithm is a region-based procedure that maximises the homogeneity within objects while minimising the average heterogeneity between objects (Watkins and van Niekerk, 2019). The procedure follows a mutual best fitting approach by starting with image objects of one pixel and repeatedly merging them based on a user-defined homogeneity criterion such as scale, shape and compactness. These parameters are combined to contribute together to the determination of a stopping threshold for intra-object homogeneity based on input layers. Consequently, this threshold defines the size and shape of the resulting image objects (Baatz and Schäpe, 2000; Benz et al., 2004).

Among the parameters used by the MRS algorithm, the scale parameter is of particular importance. Its proper selection is considered crucial, as it regulates the relative size of the image objects, which directly affects the accuracy of the results (Kim et al., 2008; Myint et al., 2011). In general, smaller values of the scale parameter result in relatively smaller image objects, while larger values result in correspondingly larger objects.

A review of the existing literature indicates the existence of a quantitative, semi automatic method for selecting optimal values for image segmentation parameters through the use of genetic algorithms. However, it is noteworthy that these semi-automated techniques have not yet been fully integrated into widely used mainstream image segmentation software (Duro et al., 2012). Nonetheless, there is no universally accepted method to identify the segmentation parameters that provide optimal pixel grouping, which implies the geographic object-based image analysis (GEOBIA) is still highly interactive and includes subjective trial-and-error methods and arbitrary decisions (Diakogiannis et al., 2020).

2.2 Instance and semantic segmentations

Within the realm of computer vision, two essential subfields have emerged as crucial components: instance segmentation and semantic segmentation.

Instance segmentation is a specialised computer vision task that not only identifies objects in an image but also distinguishes between individual instances of the same object class. Semantic segmentation, on the other hand, focuses on labeling each pixel in an image with a specific class or category, therefore resulting in a classification. Unlike instance segmentation, it doesn't differentiate between individual instances of the same object class but provides a detailed understanding of scene images. Semantic segmentation has diverse applications, from medical image analysis to environmental monitoring.

Instance segmentation can be seen as a combination of semantic segmentation and object detection, it therefore produces more meaningful inference of images and gives richer output.

2.2.1 Pixel-based vs. object-based classifications

The classification techniques can be categorised into two broad approaches for image analysis. One method of classification is based on pixels, while the other is based on objects. Pixel-based analysis has traditionally been the go-to approach for remotely sensed imagery classification, although object-based image analysis has become more prevalent (Blaschke, 2010).

Pixel-based methods operate by considering the properties of individual pixels in isolation, without considering their relationship to neighboring pixels. As a result, the initial classification in these methods is based exclusively on the spectral signature of the pixel, without taking into consideration other variables or the context of surrounding pixels (Zhou and Troy, 2008). This narrow focus on the spectral value of each pixel can lead to issues like image speckle and overall inaccuracies, especially when dealing with very high-resolution images. The speckle effect, also known as *salt-and-pepper* effect, occurs due to significant local differences in spatial patterns between neighboring pixels (Myint et al., 2011). Because each pixel is treated independently in pixel-based approaches, nearby pixels with similar appearances may be assigned different classes, contributing to the problem (Kelly et al., 2011).

Object-based image analysis (OBIA) provides a viable alternative to traditional pixel-based approaches for analysing and classifying remote sensing data from medium to high spatial resolution satellite imagery (Baatz et al., 2004; Benz et al., 2004). Indeed, numerous publications argue that object-based classification holds greater promise for categorizing higher resolution imagery compared to pixel-based methods (Oruc and Marangoz, 2004 ; Willhauck et al., 2000).

In OBIA, pixels are organised into objects or segments based on a criterion of homogeneity, either in terms of spectral or spatial characteristics. Object-based classification currently depends on the outcome of an image segmentation algorithm : in contrast to pixel-based classification methods that directly classify individual pixels, object-based classification begins by aggregating image pixels into spectrally homogeneous image objects using an image segmentation algorithm and then proceeds to classify these individual objects (Liu and Xia, 2010). The use of objects has several advantages, as these entities contain not only spectral information, such as the average band values for each object, but also various geographic and geometric features, such as the shape, length, adjacency and internal characteristics of the objects (Baatz et al., 2004). These additional attributes contribute to a knowledge base for sample objects which give far more informations than that of individual pixels. This enriched information can be used to develop a rule-based classification process (Whiteside et al., 2011).

2.2.2 Classifiers

Several classifiers have emerged as cutting-edge methods in the field of classification. The machine learning evolution provides popular non parametric classifiers like Random Forest (RF), Support Vector Machine (SVM), K-Nearest-Neighbors (KNN), convolutional neural networks (CNN), and various others. Classifiers can be categorised as either supervised or unsupervised, depending on whether they depend on labeled training samples or not. In the context of this thesis, only supervised classification has been used.

Random Forests

A RF classifier is a powerful machine learning method that combines multiple decision trees to make predictions. As shown in Niemeyer et al. (2014), it has gained popularity, particularly in the field of remote sensing, due to its robustness and high classification accuracy. Ma et al. (2017) also conducted a review of various supervised object-based classifiers and showed that Random Forest had the highest performance among the methods evaluated, with SVM closely following. However, other studies claim that

SVM outperform (Huang et al., 2002) or at least equal RF techniques (Pal, 2005).

The basic idea of a RF is to construct several decision trees using different subsets of the training data and input features, and then combine their results using a majority voting mechanism. Once the classifier has been trained on labelled data, it can be used to make predictions on new unlabelled data (Belgiu and Drăguț, 2016). Decision trees are hierarchical tree structures that represent a series of decisions based on certain variables. Each node in the tree represents a decision based on a variable, and each leaf node represents a final prediction (Belgiu and Drăguț, 2016). Decision trees are sensitive to the specific data used to train them and can suffer from overfitting and outliers. It can occur in different ensemble tree models when using Random Forests with significantly different tree structures and splitting variables. Thus, the final prediction by voting reduces overfitting when classification problems occur (Sheykhmousa et al., 2020).

Support Vector Machines

SVMs are considered to be a robust and versatile supervised non parametric learning technique that is particularly well suited to address challenges in remote sensing applications (Mountrakis et al., 2011). Most supervised learning schemes struggle due to the need for large training samples (Chi et al., 2008), whereas SVMs exhibit a remarkable ability to handle small training datasets, consistently yielding higher classification accuracy (Mantero et al., 2005).

The basic idea behind SVMs is to find an optimal hyperplane that can efficiently separate the labelled data instances in a feature space into discrete, predefined classes. This hyperplane is determined to maximise the margin, which is the distance between the hyperplane and the nearest points of each class, also known as support vectors. The maximum margin ensures better generalisation of the model to new, unlabelled data and is often referred to as the optimal separation hyperplane; it minimises misclassifications by defining a decision boundary that matches the training examples (Vapnik, 1998). The iterative learning process refines the classifier to establish an optimal decision boundary capable of separating both training samples and simulation data under the same configurations (Zhu and Blumberg, 2002).

Convolutional Neural Networks

SVM and RF methods have long been at the forefront of remote sensing applications, but in recent years considerable progress has been made in the field of CNNs (Diakogiannis et al., 2020). A CNN is a type of feed-forward neural network designed for automated feature extraction. Its basic concept involves the construction of multiple layers, each using convolutional operations, which work collectively to automatically learn hierarchical features from the data, eliminating the need for manual feature extraction. The hierarchical structure is inspired by visual perception, where artificial neurons correspond to biological neurons and each layer captures increasingly abstract representations. Activation functions within the CNN simulate the biological barrier, allowing only relevant information to pass to the next neuron above a certain threshold. Loss functions and optimisers are integral components designed by researchers to guide the CNN as it learns and refines its parameters during the training process (Li et al., 2022). This explanation is deliberately brief, as the following chapter deals with the subject in depth, as this is at the heart of this thesis.

2.3 Convolutional neural networks

CNNs are a special kind of neural network (NN) adapted to the analysis and process of images. It's part of what is called deep learning, which is a branch of machine learning that has proven valuable for solving difficult problems such as recognising objects in images or language. In recent years, the use of such neural networks has become popular in a variety of application domains. The reason for this enthusiasm is that image classification with deep neural networks has been improving steadily over the years until it has produced promising results. The research of Krizhevsky et al. (2001) marked a considerable turning point with AlexNet, an 8-layer CNN that won the ImageNet challenge in 2012. This network demonstrated that the crucial point for making big networks to work is the availability of large training sets (Theodoridis, 2020). Important research papers have since shown even larger and deeper neural networks (He et al., 2016 ; He et al., 2020).

Unless otherwise stated, all the general theory related to neural networks presented in this section comes from the course *CS231n : Convolutional Neural Networks for Visual Recognition* (Karpathy, 2016) given at Stanford University. The theory related to the ResUNet-a model is taken mainly from studies conducted by Ronneberger et al. (2015), Diakogiannis et al. (2020) and Waldner and Diakogiannis (2020).

2.3.1 How a neural network works

Neural networks, inspired by the structure and function of the human brain, are computational algorithms designed for pattern recognition. They excel at learning intricate relationships between specific features and making generalisations, inferences and associations that may elude human perception.

The human brain is a vast network of hundreds of millions of neurons connected by synapses. Activation of a neuron occurs when a critical mass of synapses connected to it is stimulated, resulting in the generation of an outgoing signal via output synapses. The intricate connectivity between these millions of neurons and synapses fundamentally shapes cognitive processes and behaviour. In artificial neural networks, biological neurons find their counterparts in artificial neurons whose connections are established through synapses with adjustable weights.

The activation of an artificial neuron depends on two key factors:

- **Weights:** These represent the synaptic strengths used to combine inputs to the neuron. These weights are dynamically adjusted as the neural network learns.
- **Activation function:** This is the computational operation that determines the response of the artificial neuron. It can be as simple as a threshold function that defines the conditions under which neurons are activated.

Together, these neurons form different layers within the neural network architecture. Within each layer, a given neuron can be modulated by neurons in the previous layer, using their outputs as inputs. This inter-layer connectivity allows information to propagate through the network, facilitating the complex processing and transformation of input data across successive hierarchical levels. When all the inputs of a layer are connected to all the nodes of the next layer, the neural network is said to be fully connected, as shown in Figure 2.1. Other neural network models exist such as convolutional and recurrent neural networks, which have the advantage of being easier to train due to the smaller number of neurons.

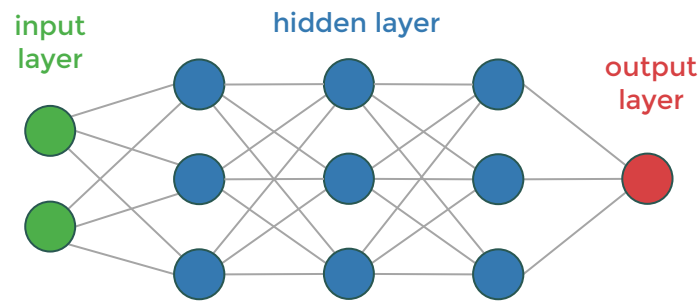


FIGURE 2.1: Example of fully connected neural network

The layered configuration of neural networks is designed to facilitate data clustering and classification. The NN operates by iteratively adjusting its internal parameters, in particular the weights of the neurons, through a training process. This adjustment is guided by the minimisation of a specified loss function as explained below.

2.3.2 Key-concepts of neural networks

Activation function

The activation function is a mathematical expression that determines the output of a neuron within a neural network. It serves as a crucial component in deciding whether a neuron is activated or remains inactive and thus, if the input signals is transmitted to the next layer. Modern neural networks widely use nonlinear activation functions because they facilitate complex data learning and accurate prediction. Figure 2.2 illustrates two of the most commonly employed activation functions.

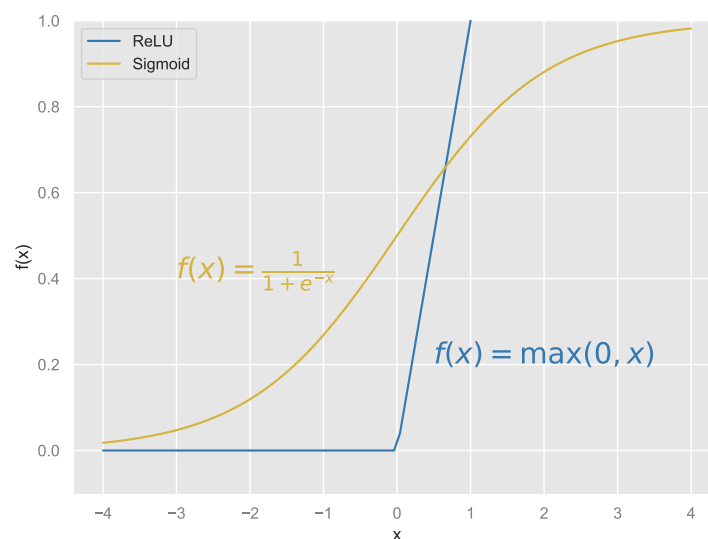


FIGURE 2.2: Sigmoid and ReLU : two of the most commonly employed activation functions

The sigmoid function is often used because of its computational simplicity and its ability to constrain output values, preventing them from becoming too large. However, a notable limitation of classical activation functions, including the sigmoid, is their susceptibility to the vanishing gradient problem. This problem consists of exponentially decreasing gradients during backpropagation, which prevents the effective learning of early layers in deep neural networks and hinders the ability of the model to capture complex patterns.

The Rectified Linear Unit (ReLU) is an alternative activation function that consists in replacing all negative values by zeros. By maintaining only the positive values, only the activated features are transmitted to the next layer. It has been reported that in the context of deep networks, the use of ReLU nonlinearity in the hidden layers can significantly speed up the training time (Theodoridis, 2020). It allows to generalise the learning and diminish the noise within the network.

Loss function

The loss function is a mathematical function that measures the difference between the predicted output and the actual output for a given input data sample. A loss function is also referred as a cost function. The goal of the loss function is to provide a quantitative measure of how well the model is performing in predicting the correct output, it is used to guide the optimisation process by providing feedback on how well it fits the data.

Gradient descent

Gradient descent, on the other hand, is an optimisation algorithm used to minimise the loss function. The idea behind gradient descent is to iteratively update the model parameters in the direction of steepest descent of the loss function, until a minimum is reached. In other words, gradient descent is a process of finding the optimal values of the model parameters that result in the smallest possible value of the loss function.

When adopting a gradient descent scheme to minimise a nonconvex cost function, the algorithm can converge to either a local or to a global minimum. However, this is not necessarily bad news, if the local minimum is deep enough, convergence to such a local minimum can correspond to a good solution (Theodoridis, 2020).

Back propagation

Backpropagation involves a forward pass where input data is processed through the network to generate predictions, followed by the calculation of the error between these

predictions and the actual target output using a specified loss function. The error is then propagated backwards through the network and the weights and biases are updated using gradient descent to minimise the error. This iterative process is repeated until the model converges on a set of parameters that produce accurate predictions on the training data. Backpropagation enables neural networks to learn and adapt, and plays a crucial role in optimising deep learning models.

2.3.3 Typical architecture of a convolutional neural network

A CNN differs from a classical neural network in its specialized architecture tailored for processing images with spatial arrangements. A CNN is built with two main types of blocks: convolutional and pooling layers (Figure 2.3).

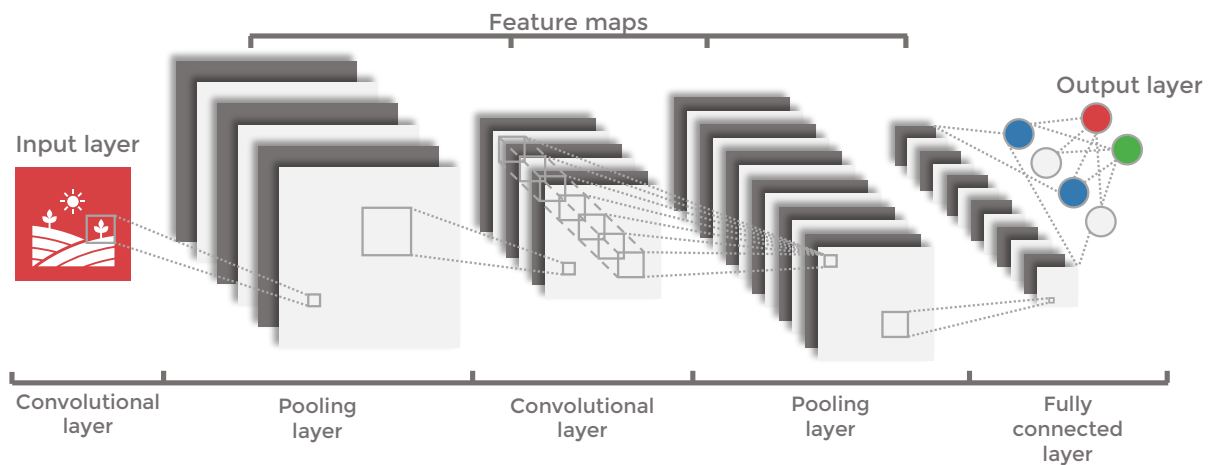


FIGURE 2.3: Standard CNN architecture

Convolutional layer

The convolutional layer is the most important building block of a CNN. This layer allows the extraction of image features by the application of filters.

The input image is composed of pixels with an associated value in order to obtain a matrix. This matrix is then multiplied by a filter, also a matrix, which moves from pixels to pixels from left to right and from top to bottom until the image has been completely crossed (Figure 2.4).

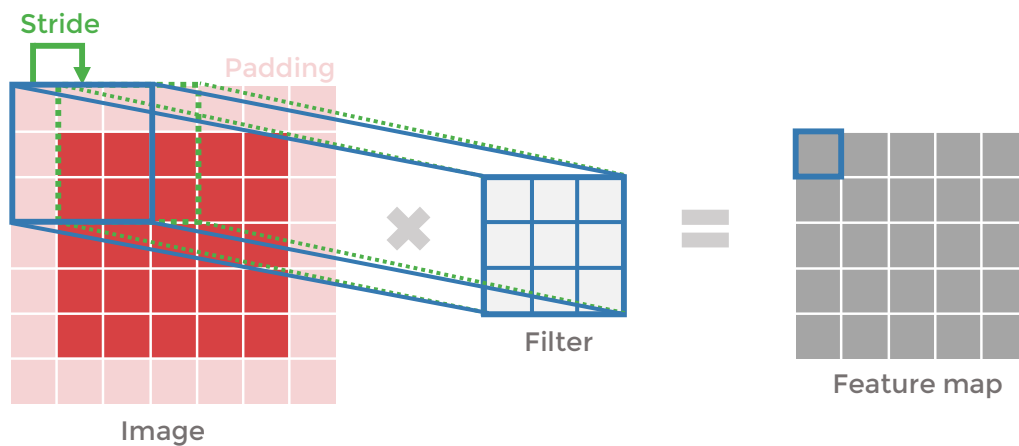


FIGURE 2.4: Standard representation of a convolution matrix with stride and padding hyperparameters

The result of the multiplication of each filter's cell is summed and divided by its total number of cells. This final smaller matrix is called a feature map. The continuous sliding of the filter across the width and height of the input image is called a convolution, in fact the filter *convolves* across the image. The output of this operation is a single value that represents the activation of the corresponding neuron in the output feature map.

During the training process, the CNN learns to adjust the values of the filter weights in order to extract useful features from the input images. The filters are learned by minimizing a loss function that measures the difference between the predicted output of the CNN and the true labels of the training data. This is typically done using back-propagation and gradient descent optimization.

The filter is not set manually, the convolution layer learns by itself during training which filters are most useful to perform its task. Moreover, the layers above will learn to combine them into more complex patterns (Géron, 2019).

Some spatial arrangements can be programmed by the user. These options to control the convolution layer are commonly called hyperparameters and are illustrated in Figure 2.4:

- **The stride :** The stepover achieved by the filter can be specified. Indeed, when the stride is 1, it means that the filter slides one pixel at a time. The larger the step size, the smaller the output feature map will be. In other words, this parameter enables to control the amount of volume that will be output.

- **The zero padding :** In order for a layer to have the same height and width as the previous layer, it is common to add zeros around the inputs. This hyperparameter can be convenient in some situations. No padding leads to a size reduction of the output volume.

Pooling layer

The function of the pooling layer is to reduce the computational load, the memory usage and the number of parameters that the network needs to learn. A pooling neuron has no weight, all it does is aggregate the inputs using an aggregation function such as the max or mean (Géron, 2019).

Like the filter in the convolution layer, a sliding window moves along the input and selects the largest value (in the case of max pooling), this value is the only one in the considered window to be propagated to the next layer while the others are dropped (Figure 2.5). The stride, the size and the padding can be adjust such as a convolutional layer.



FIGURE 2.5: Example of max pooling

2.3.4 Description of ResUNet-a architecture

A U-Net backbone architecture

The ResUNet-a model is a deep convolutional neural network that uses a U-Net encoder/decoder backbone. Like any convolutional neural network, the structure of U-Net consists of a multitude of mathematical operations illustrated in Figure 2.6.

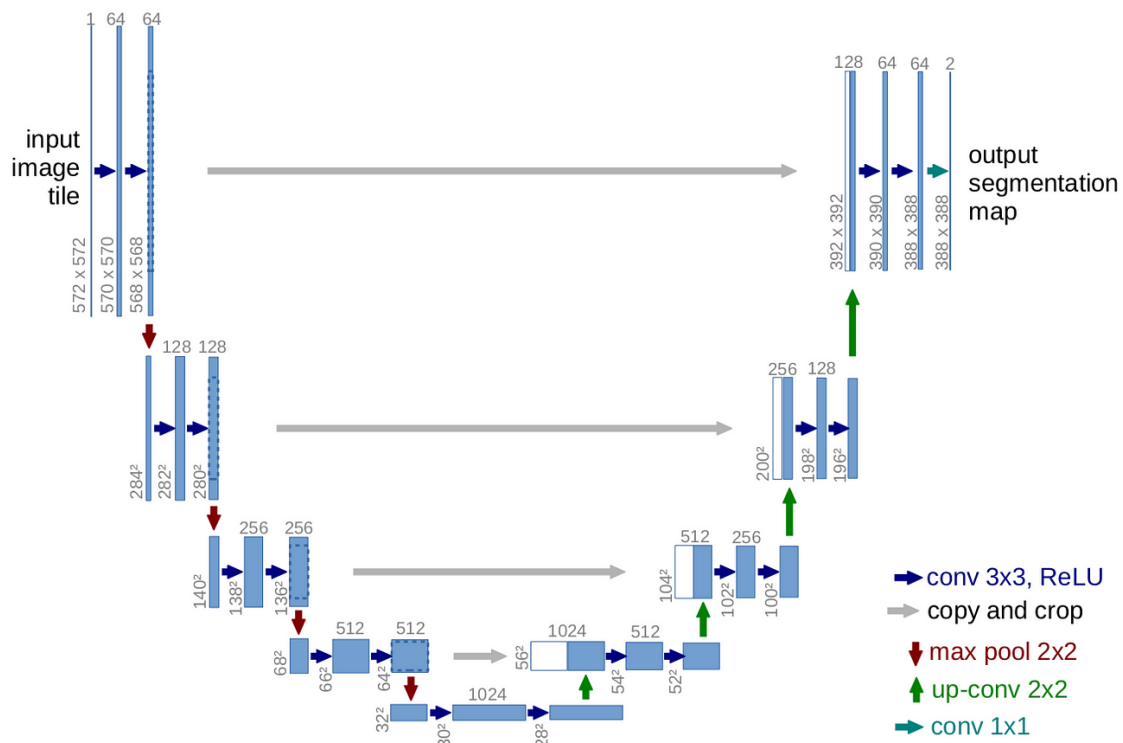


FIGURE 2.6: U-Net architecture (Ronneberger et al., 2015)

The descending segment of the U-shaped structure compresses the data by reducing the spatial dimensions, allowing the network to capture complex features through the repeated application of two 3x3 convolutions (unpadded), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with a stride of 2 for downsampling. At the same time, the number of feature channels is doubled at each downsampling step (Ronneberger et al., 2015).

In contrast, the ascending part of the U-shaped structure involves expanding the data to restore the original spatial dimensions of the input, resulting in a high-resolution segmentation map. This process includes steps such as upsampling the feature map, a 2x2 convolution that halves the feature channels, concatenation with the cropped feature map from the contraction path, and application of two 3x3 convolutions with ReLU activation at each step. Cropping is essential due to the loss of boundary pixels in each convolution. The final layer uses a 1x1 convolution to map each 64-component feature vector to the desired number of classes, summing to a network with a total of 23 convolutional layers (Ronneberger et al., 2015).

ResUNet-a architecture

The difference between U-Net and ResUNet-a lie in the fact that the second one incorporates residual blocks (He et al., 2016), atrous convolutions (Diakogiannis et al., 2020), pyramid scene parsing pooling (Zhao et al., 2017) and multi-tasking inference (Diakogiannis et al., 2020).

As proposed by He et al. (2016), the UNet architecture's building blocks are replaced with modified residual blocks of convolutional layers. These residual blocks play a crucial role in mitigating the challenges associated with vanishing and exploding gradients in deep architectures.

To enhance understanding across different scales, each residual building block also incorporates multiple parallel atrous convolutions with varying dilation rates. The inclusion of these diverse-scale layers aims to capture object features at various receptive field scales, with the expectation that this strategy will improve performance by recognising correlations among objects at different locations in the image (Diakogiannis et al., 2020).

Moreover, the integration of the pyramid scene parsing (PSP) pooling layer further enhances the network's performance by incorporating background context information (Zhao et al., 2017).

Lastly, the ResUNet-a model offers two variations, ResUNet-a d6 and ResUNet-a d7, allowing multi-task inference. These variations differ in their depth, determined by the number of layers (Diakogiannis et al., 2020). ResUNet-a d6 has six residual building blocks in the encoder followed by a PSP Pooling layer while ResUNet-a d7 has seven building blocks instead. The ResUNet-a model, as used by Diakogiannis et al. (2020) in the context of field delineation, is set up to produce four different tasks : 1) a map of the probability of each pixel to be in a field; 2) a map of the probability of each pixel to be a boundary; 3) a map of the distance to the closest boundary; 4) a reconstruction of the input image in the Hue-Saturation-Value (HSV) color space. The architecture of the model is shown in Figure 2.7.

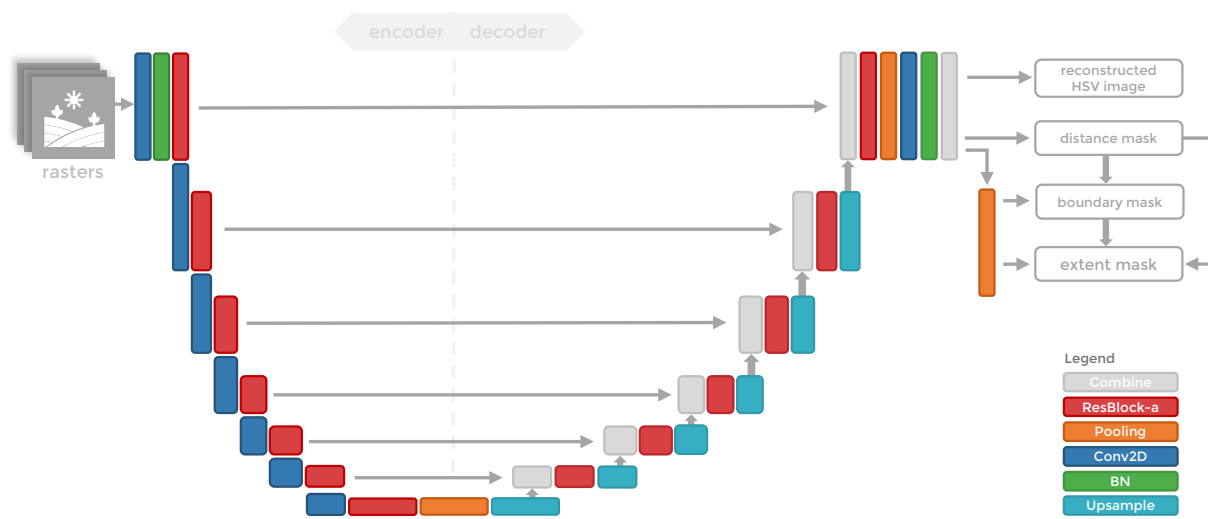


FIGURE 2.7: The ResUNet-a architecture used in the context of field delineation

The different types of layers, forming together the architecture of ResUNet-a, can be summarized as:

Conv2D layer : represents a standard 2D convolutional layer featuring a kernel size of 3 and a padding of 1.

Batch Normalisation (BN) layer : represents a 2D convolution layer, featuring a kernel size of 3 and a padding of 1, followed by a batch normalisation. A BN is a technique that significantly enhances the speed, performance, and stability of artificial neural networks by subtracting the batch mean of the output of a previous activation layer, and dividing it by the batch standard deviation (Ioffe and Szegedy, 2015).

ResBlock-a layer : is a residual unit that facilitates the transmission of information from the layers in the contracting path to the corresponding segments of the expanding path (He et al., 2016). In contrast to standard residual units, characterized by a single branch comprising two consecutive convolution layers, ResBlock-a features four parallel branches of atrous convolutions with progressively higher dilatation rates. These branches enable the processing of information at various scales (Waldner and Diakogiannis, 2020).

PSP Pooling layer : accentuates contextual information through the application of maximum pooling at four distinct scales (Zhao et al., 2017). The initial scale involves global max pooling. In the second scale, the feature map is partitioned into four equal

areas, and max pooling is applied within each of these areas, a pattern repeated for the subsequent two scales. The multiple scales are respectively 256, 128, 64 and 32 in the ResUNet-a model.

Upsample layer : is a layer in the expanding segment that doubles the size of the feature map while the number of filters is halved. The upsampling is done by interpolation and is followed by a Conv2D layer and a batch normalisation.

Combine layer : is a layer that receives two inputs, each featuring the same number of filters in every feature map. It concatenates these inputs and generates an output at the identical scale, maintaining the same number of filters as each of the input feature maps.

Output layer : performs multiple tasks with conditioned inference. Four correlated tasks are produced by the model in the study of Diakogiannis et al. (2020) : field extent map, field boundaries map, distance to the nearest boundary and reconstruction of the input image in HSV color space (see section 4.6.3).

Loss function

The loss function used in the ResUNet-a model is the Tanimoto loss and is based on the DICE coefficient. It's defined as the average of the Tanimoto distance $T(\mathbf{p}, \mathbf{l})$ and its complement $T(\mathbf{1} - \mathbf{p}, \mathbf{1} - \mathbf{l})$:

$$\tilde{T}(\mathbf{1} - \mathbf{p}, \mathbf{1} - \mathbf{l}) = \frac{T(\mathbf{p}, \mathbf{l}) + T(\mathbf{1} - \mathbf{p}, \mathbf{1} - \mathbf{l})}{2} \quad (2.1)$$

with

$$T(\mathbf{p}, \mathbf{l}) = \frac{\sum_i p_i l_i}{\sum_i (p_i^2 + l_i^2) - \sum_i (p_i l_i)} \quad (2.2)$$

where $p \equiv p_i \in [0; 1]$ represents the vector of probabilities for the i th pixel, and l_i are the corresponding ground truth labels (Waldner and Diakogiannis, 2020).

For multiple segmentation tasks, the complete loss function is defined as the average of the loss of all tasks:

$$\tilde{T}_{\text{MTL}}(\mathbf{p}, \mathbf{l}) = 0.25 [\tilde{T}_{\text{extent}}(\mathbf{p}, \mathbf{l}) + \tilde{T}_{\text{boundary}}(\mathbf{p}, \mathbf{l}) + \tilde{T}_{\text{distance}}(\mathbf{p}, \mathbf{l}) + \tilde{T}_{\text{reconstruction}}(\mathbf{p}, \mathbf{l})] \quad (2.3)$$

Waldner and Diakogiannis (2020) demonstrated in their study that the Tanimoto loss outperforms the standard Dice or the cross-entropy losses. The Tanimoto distance with complement maintains an equilibrium between tasks and accelerates the training convergence regardless of weights initialisation.

2.4 Image segmentation evaluation metrics

In the context of image segmentation and object detection, it's crucial to understand key concepts like True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). These concepts help in evaluating the performance of models by analysing the predictions in relation to the ground-truth bounding boxes.

True Positive (TP) refers to instances where both the predicted and actual values are positive. In object detection, this signifies a correct detection of a ground-truth bounding box, which represents the annotated region corresponding to the target object. Conversely, False Positive (FP) occurs when the predicted value is positive, but the actual value is negative. In object detection, this implies the model has detected the presence of an object that does not exist or has localized it inaccurately. False Negative (FN) is the case where the predicted value is negative, but the actual value is positive. In object detection, this signifies a failure to detect a ground-truth bounding box despite its actual presence. True Negative (TN) corresponds to instances where both predicted and actual values are negative. In the context of object detection, TN might represent areas correctly identified as not containing an object. These concepts can be represented in a confusion matrix (Figure 2.8).

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

FIGURE 2.8: Confusion matrix

2.4.1 Pixel-based metrics

Precision and Recall

Precision and *Recall* are specific measures of the performance of a model in the context of classification, image segmentation and object detection. They are often applied at the level of instances, objects, or categories in various computer vision tasks.

Precision is a metric that quantifies the accuracy of positive predictions made by a model. Precision is calculated as the ratio of true positives to the sum of true positives and false positives. A high precision indicates that when the model predicts the presence of an object, it is likely to be correct, which is particularly important when false positives are costly or undesirable. Precision can be written as :

$$Precision = \frac{TP}{TP + FP} \quad (2.4)$$

Recall, in the other hand, measures the ability of a model to capture all the relevant instances of a class. It is calculated as the ratio of true positives to the sum of true positives and false negatives. A high recall indicates that the model is effective at finding most of the actual positive instances. Recall can be written as :

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

Overall Accuracy

The overall accuracy (OA) is a way of measuring how well a classification model performs. It's the most commonly used metric for classification models. Calculating the OA consists to take the number of pixels that were predicted correctly by the model and divide it by the total number of pixels. This gives a number between 0 and 1, where 1 means that the model made no mistakes and correctly classified every pixel. The OA is defined by

$$OverallAccuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.6)$$

This metric should be used in conjunction with other metrics to get a more comprehensive understanding of the model's performance. The OA does not distinguish between types of error, and its result may also vary if the classes analysed are unbalanced. If a majority class is always predicted with high accuracy, the overall accuracy score will

be high, whereas the model does not necessarily perform well for minority classes. Explained in another way, the OA can have a high accuracy even if the precision or recall has a low value. The use of other metrics therefore enables a deeper understanding of the model by better identifying its errors and taking into account the different biases associated with this single metric.

F1-Score

In many scenarios, a trade-off exists between precision and recall. For a more balanced evaluation, the F1-Score, which is the harmonic mean of precision and recall, can be employed. The F1-Score, or DICE score, provides a consolidated metric that considers both precision and recall, making it particularly useful in situations where there is an imbalanced dataset. The F1-score is defined by

$$DICE = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (2.7)$$

Such as the OA, the DICE score ranges from 0 to 1, where 1 means that the model made perfect predictions.

Matthews Correlation Coefficient

The Matthews correlation coefficient (MCC) is a metric that indicates the correlation between the predictions and the validation data. It stands out as the most reliable pixel metric for evaluating binary segmentation. In their study, Chicco and Jurman (2020) assess that the MCC provides a high score only when predictions show favourable results in all four categories of the confusion matrix (Figure 2.8). This score is proportional to both the size of the positive elements and the size of the negative elements within the dataset. The MCC is defined by

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (2.8)$$

and ranges from -1 to 1, where 1 means that the model made perfect correlation.

2.4.2 Object-based metrics

Intersection over Union

The Intersection over Union (IoU) metric is based on the Jaccard Index, a coefficient for gauging the similarity of datasets. The IoU is used to evaluate the performance of object detection algorithms by measuring the overlapping area between a predicted

bounding box B_p and a ground-truth bounding box B_a divided by the area of union between them for a given object (Padilla et al., 2020), which is defined by

$$IoU(B_p, B_a) = \frac{area(B_p \cap B_a)}{area(B_p \cup B_a)} \quad (2.9)$$

The IoU metric is commonly used as a measure of the accuracy of an object detection algorithm. A higher IoU indicates a better match between the predicted bounding box and the ground-truth bounding box as shown in Figure 2.9. In practice, a threshold value t of the IoU is often set to determine whether a predicted bounding box is considered a true positive or a false positive. The IoU value needs to be greater or equal to the threshold t to consider the detection of the given object as true positive, otherwise it's considered as false positive (Mahdi et al., 2020). The value of IoU is always between 0 and 1. By convention, a predicted bounding box B_p is considered as being reasonably good if $IoU(B_p, B_a) \geq 0.5$ (Ng, 2018).

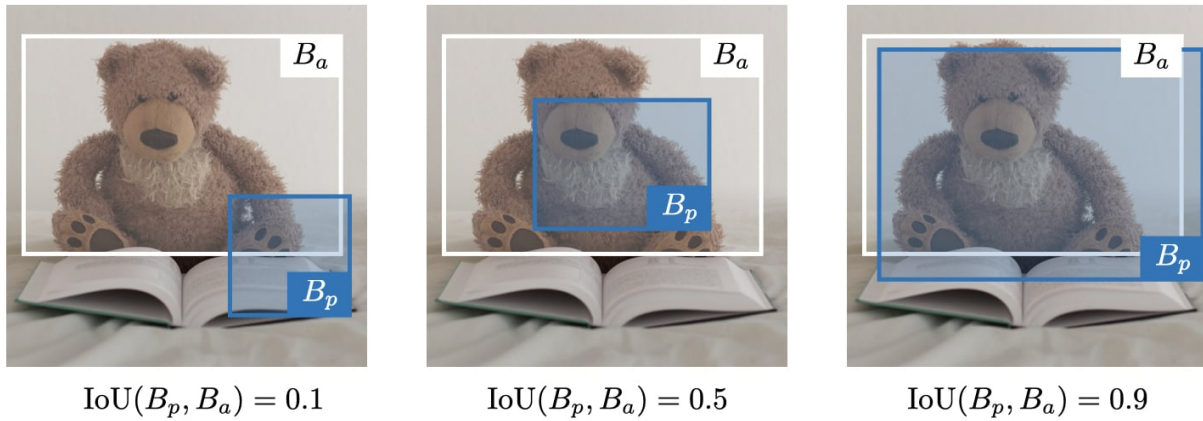


FIGURE 2.9: Illustration of the predicted bounding box B_p and the ground-truth bounding box B_a associated with the IoU score (Amidi and Amidi, 2019)

For the purposes of this thesis, the IoU metric is derived into two different metrics: the IoU_{median} and the IoU_{50} . The IoU_{median} is obtained by taking the median of the IoUs while IoU_{50} is the number of objects whose IoU is greater than 50% divided by the number of objects in the reference dataset.

RecRate_{soft}

Based on the Jaccard index, like the IoU, Pavlova et al. (2021) have introduced the RecRate_{soft} metric which, unlike the other metrics explained above, does not suffer from over- and under-segmentation errors. Initially, the RecRate index consists of calculating a one-to-one correspondence between two objects f_{ref}^i and f_{pred}^j , the first coming from a reference dataset and the other from the prediction of a deep learning model.

Under the conditions that $J(f_{\text{ref}}^i, f_{\text{pred}}^j) > 0.5$ and that these objects do not overlap, the objects f_{ref}^i and f_{pred}^j are considered to correspond to each other. The RecRate index is thus defined by

$$\text{RecRate} = 2 \cdot \frac{N_{\text{one2one}}}{N_{\text{exp}} + N_{\text{ref}}} \cdot 100\% \quad (2.10)$$

where N_{one2one} corresponds to the number of one-to-one matches found between f_{ref}^i and f_{pred}^j elements, N_{ref} and N_{pred} corresponding to the total number of objects in the reference and prediction datasets.

The RecRate index is sensitive to false positives and false negatives, which is already an improvement on the IoU metric given that the latter does not take into account the number of false positives. However, the RecRate index is sensitive to over- and under-segmentation and penalizes for merge/split errors. The $\text{RecRate}_{\text{soft}}$ counters this drawback as it does not take only the one-to-one correspondences between the elements f_{ref}^i and f_{pred}^j into account. $\text{RecRate}_{\text{soft}}$ is defined by the following criterion:

$$\Omega(a, b) = \max \left\{ \frac{S(a \cap b)}{S(a)}, \frac{S(a \cap b)}{S(b)} \right\} \quad (2.11)$$

which generates a set of correspondences between the objects in the reference and prediction sets.

The f_{ref}^i object corresponds to the f_{pred}^j object if both of the following conditions are true:

$$f_{\text{pred}}^j = \underset{f \in f_{\text{pred}}}{\operatorname{argmax}} \left(\Omega \left(f_{\text{ref}}^i, f \right) \right) \quad (2.12)$$

$$\Omega \left(f_{\text{ref}}^i, f_{\text{pred}}^j \right) \geq 0.5 \quad (2.13)$$

Conversely, the f_{pred}^j object corresponds to the f_{ref}^i object if the following two conditions are true:

$$f_{\text{ref}}^i = \underset{f \in f_{\text{ref}}}{\operatorname{argmax}} \left(\Omega \left(f_{\text{pred}}^j, f \right) \right) \quad (2.14)$$

$$\Omega \left(f_{\text{pred}}^j, f_{\text{ref}}^i \right) \geq 0.5 \quad (2.15)$$

It allows multiple objects in the reference dataset to correspond to an object in the prediction, and multiple objects in the prediction to correspond to an object in the reference data, thus ignoring over- and under-segmentation errors. The $\text{RecRate}_{\text{soft}}$ is

therefore finally defined by

$$RecRate_{soft} = \frac{N_{ref}^{\Omega}}{N_{ref}} \cdot 100\% \quad (2.16)$$

where N_{ref}^{Ω} corresponds to the number of objects in the reference set that are part of the matching set, denoted F_{Ω} , and where N_{ref} corresponds to the total number of objects in the reference set. Comparing the differences in results between $RecRate_{soft}$ and IoU shows the importance of over- and under-segmentation errors.

2.5 Related work

The use of deep learning in the context of remote sensing has attracted significant interest in the latest years. Various studies have been conducted with CNN for the detection of agricultural fields, but also for other applications (Brodrick et al., 2019).

A first study, freed from advances in deep learning and conducted by Watkins and van Niekerk (2019), compares and evaluates multiple OBIA techniques for automatically delineating agricultural field boundaries, using multi-temporal Sentinel-2 imagery. The authors combine two edge detection algorithms (Canny and Scharr) and three image segmentation algorithms (watershed, multi-resolution, and multi-threshold) to conduct six experiments. The results show that the watershed segmentation (WS), combined with Canny edge detection, produces the most accurate field boundaries. The Canny edge detection algorithm performs better than the Scharr algorithm, providing finer field boundary edges. The WS is found to be more suitable than multi-resolution segmentation (MRS) and multi-threshold segmentation for delineating agricultural field boundaries. The study concludes that the multi-temporal approach, using the combination of Canny edge detection and WS, holds potential for accurately delineating field boundaries, and acknowledges the limitations of the Sentinel-2 imagery's relatively low spatial resolution for mapping small, narrow, or irregularly shaped fields. The research findings suggest that applying the Canny edge detection algorithm and WS on a larger scale in other agricultural environments could yield favorable outcomes.

Another relevant study has been conducted by Diakogiannis et al. (2020) concerning the delineation of field parcels located in South Africa, this time using deep learning. The authors used the model ResUNet-a to analyse satellite images and identify three different outputs: the extend of fields, the field boundaries and the distance to the closest boundary. They observed that the model's performance and its ability to

generalise greatly improve by asking the algorithm to reconstruct the three correlated outputs precited. They first used a single monthly composite image from Sentinel-2 as input and obtained highly accurate results on individual fields identification, but only by achieving post-processing techniques such as thresholding or watershed segmentation. In a second time, the researchers discovered that using multiple images taken over a season helped to reduce the temporal variations of accuracy without recalibration. The ResUNet-a model has been shown to accurately detect field boundaries and discard irrelevant boundaries, and thereby outperforming conventional edge filters. For South African fields averaging 13 ha, the average accuracy (F1-Score) is 92%. Moreover, 99% of the fields used during training are well identified, both in terms of their shape and position.

Another research conducted by Waldner et al. (2021) presents a method, DECODE (DEtect, COnsolidate, and DELinetate), which automatically extracts accurate field boundary data from satellite imagery using deep learning based on spatial, spectral, and temporal indices. The researchers introduce a new CNN, FracTAL-ResUNet, and two uncertainty metrics to characterize the confidence of the field detection and field delineation processes. The study demonstrates the performance and scalability of the method by extracting fields across the Australian grains zone with an overall pixel-based accuracy of 87% and an overall field-based accuracy of up to 88%. Additionally, the authors trained a model on data from South Africa and found that it transferred well to unseen Australian landscapes, indicating the operational maturity of large-scale field boundary extraction based on deep learning.

Still using the FracTAL-ResUNet model, a study conducted by Wang et al. (2022a) explores the challenges and solutions for automated field delineation in smallholder agricultural systems. The authors addresses the lack of field boundary datasets in smallholder systems due to small fields requiring high-resolution satellite imagery and a lack of ground labels for model training and validation. The study aims to overcome these challenges by combining transfer learning and weak supervision and demonstrates the success of these methods in India by generating efficiently 10,000 new field labels. The researchers used a state-of-the-art neural network pre-trained on France field boundaries and then fine-tuned on India labels, achieving a median Intersection over Union (IoU) of 86% in India. They also experimentally show that pre-training in France reduces the number of India field labels needed to achieve a given performance level by as much as 20x when datasets are small. The study found that training on partial labels is preferable when labeling resources are constrained and that multi-temporal but lower-resolution PlanetScope imagery outperformed the consensus of

multiple months of imagery in India. The findings of this work suggest that by efficiently collecting partial labels and using transfer learning, smallholder fields can be delineated at large geographic scales.

Previous research has also explored the use of fully convolutional networks and combinatorial grouping to map agricultural parcels in smallholder farms (Persello et al., 2019). Two study areas were analysed, in Nigeria and Mali. Accurately mapping agricultural resources in these regions can be difficult due to the small plot size (< 2 ha), irregular field shapes with unclear boundaries, and seasonal variations in surface reflectance. The researchers used very high resolution (VHR) satellite imagery and employed the SegNet model, a convolutional neural network that can learn complex features in the image and identify sparse field contours accurately. A hierarchical segmentation was derived from the contours using the oriented watershed transform and by iteratively merging adjacent regions based on the average strength of their common boundary. Finally, field segments were obtained by adopting a combinatorial grouping algorithm exploiting the information of the segmentation hierarchy. Their proposed method shows promising results, delineating field boundaries automatically with F-scores higher than 70% and 60% in the two study areas in Nigeria and Mali, respectively, thereby outperforming alternative methods.

In another work, Jong et al. (2022) also proposed a new approach, this time using an adversarial training framework to improve field boundary prediction. They trained a ResUNet model as a generator in a traditional generative adversarial network (GAN) setup to induce a "well-formed boundary style" on the distribution of model outputs. The results showed that adversarial training substantially improved boundary quality and performance, particularly by reducing the need for post-processing methods. However, the transfer of the trained model to label-scarce regions in Brazil showed limited success, in contrary to Waldner et al. (2021).

In conclusion, the different studies presented highlight the progress and potential of deep learning in remote sensing for agricultural land delineation, with different methodologies and models showing promising results. These promising results are one of the arguments for choosing the ResUNet-a model to answer the question posed in this thesis. Since then, other newer and more promising models have emerged, such as the FracTAL-ResUNet model, but ResUNet-a has a less complex structure and therefore has the advantage of being more accessible in the context of this thesis.

Chapter 3

Objectives

Numerous studies have shown that deep learning models excel at image segmentation and classification, outperforming more traditional methods. The use of such deep learning models for the delineation and extraction of agricultural parcel boundaries makes no exception. In this domain, Waldner and Diakogiannis (2020), among others, have shown very satisfactory results in evaluating the performance of a convolutional neural network called ResUNet-a. However, the problem remains that these advanced models require a very large amount of training data to excel at their task. For field boundary delineation, the lack of field data to train the ResUNet-a model can quickly become a limiting factor in addition to the associated computational cost. The main objective of this research is to assess whether it is possible to partially replace field data with data obtained using traditional image segmentation algorithms, which do not require field data as input, without affecting the performance of the ResUNet-a convolutional neural network. This research therefore seeks to answer the question:

To what extent are field data dispensable for training the ResUNet-a neural network in automatic field boundaries delineation ?

To answer this question, several objectives are pursued.

The first one is to preprocess the images that make up the dataset for training and validating the ResUNet-a model and on which an image segmentation algorithm is executed. The preprocessing includes three steps : cloud mask application, NDVI calculation and composites aggregation. The images used come from the Sentinel-2 satellite and focus on Wallonia in Belgium, as access to existing data is a prerequisite for this work. However, part of the ambition is to apply this deep learning model to regions of the world where data is absent or scarce.

The second objective is to apply a multi-resolution segmentation on the pre-processed images. The aim of this phase is to obtain a segmentation of existing agricultural

parcels using the MRS algorithm via the eCognition software, as opposed to the segmentation of agricultural parcels obtained manually by farmers. To improve segmentation, a SVM classification is also applied to differentiate the field class from the non-field class.

The third objective focuses on the ResUNet-a convolutional neural network and is composed of several steps, including training, inference and instance segmentation among others. Multiple models are trained. Some is trained from scratch, i.e without transferring the learning of a pre-trained model, with different percentages of field data. These trained models from scratch corresponds to a supervised training, while other models, named mix, are first pre-trained with data obtained by an algorithm, corresponding to self-supervised training, and then trained with different percentages of field data.

The final objective is therefore to analyse the performance of the different models, and requires the use of image segmentation evaluation metrics selected in the literature review. The evaluation of the model performance consists precisely in quantifying the volume of field data that can be dispensed with for the training of the model without reducing its capacity to delineate field boundaries.

Chapter 4

Material and methods

This chapter details the comprehensive process of data collection and experimentation undertaken to address the central question posed in this thesis. The section 4.1 presents the chosen study area followed by the time period (4.2), while the section 4.3 presents the different datasets collected and their characteristics. The section 4.4 covers all the preprocessing techniques used to generate the different inputs required for the eCognition segmentation and for the training and evaluation of the ResUNet-a model. Finally, while the penultimate section (4.5) explains the processes used for the eCognition segmentation, the last one (4.6) covers the entire methodology specific to the ResUNet-a model, from its training to the evaluation of its performance.

4.1 Study area

The study area, located in Wallonia in the south of Belgium, is intended to cover the diversity of agricultural fields present in Wallonia. Wallonia has a temperate oceanic climate characterized by moderate temperatures, abundant cloud cover and frequent precipitation. The study area includes three distinct training zones and an inference zone (Figure 4.1). The interest is to teach the convolutional neural network to recognize all the features of field plots which have particularities according to the region.

4.1.1 Brabant and Hennuyer plateaus

The Hennuyer plateau and the Brabant plateau are characterized by lowlands with large fertile areas. These two regions can be generalized as being the silty plateaus of Wallonia even if the Hennuyer region presents some more sandy-silty zones. Such a relatively flat surface as well as the agricultural richness of the soils of the silty region favored the intensive production of large-scale crops. The average surface area of fields in this region is 2.41 hectares.

4.1.2 Hesbaye

The region of Hesbaye is very fertile with soils rich in silt and clay which make them well drained. The Hesbaye is a rather flat region particularly known for its vast cultivated fields as far as the eye can see, they are also often called *openfields*. The average surface area of fields in this region is 2.99 hectares.

4.1.3 Ardennes

The Ardennes region is very hilly, its relief has been shaped by the Meuse River and numerous rivers. The Ardennes are known for the important presence of forest massifs and its grasslands on steep grounds. Its significant topography makes the Ardennes more of an extensive agricultural region, and rather based on livestock and forestry. The average surface area of fields in this region is 2.33 hectares.

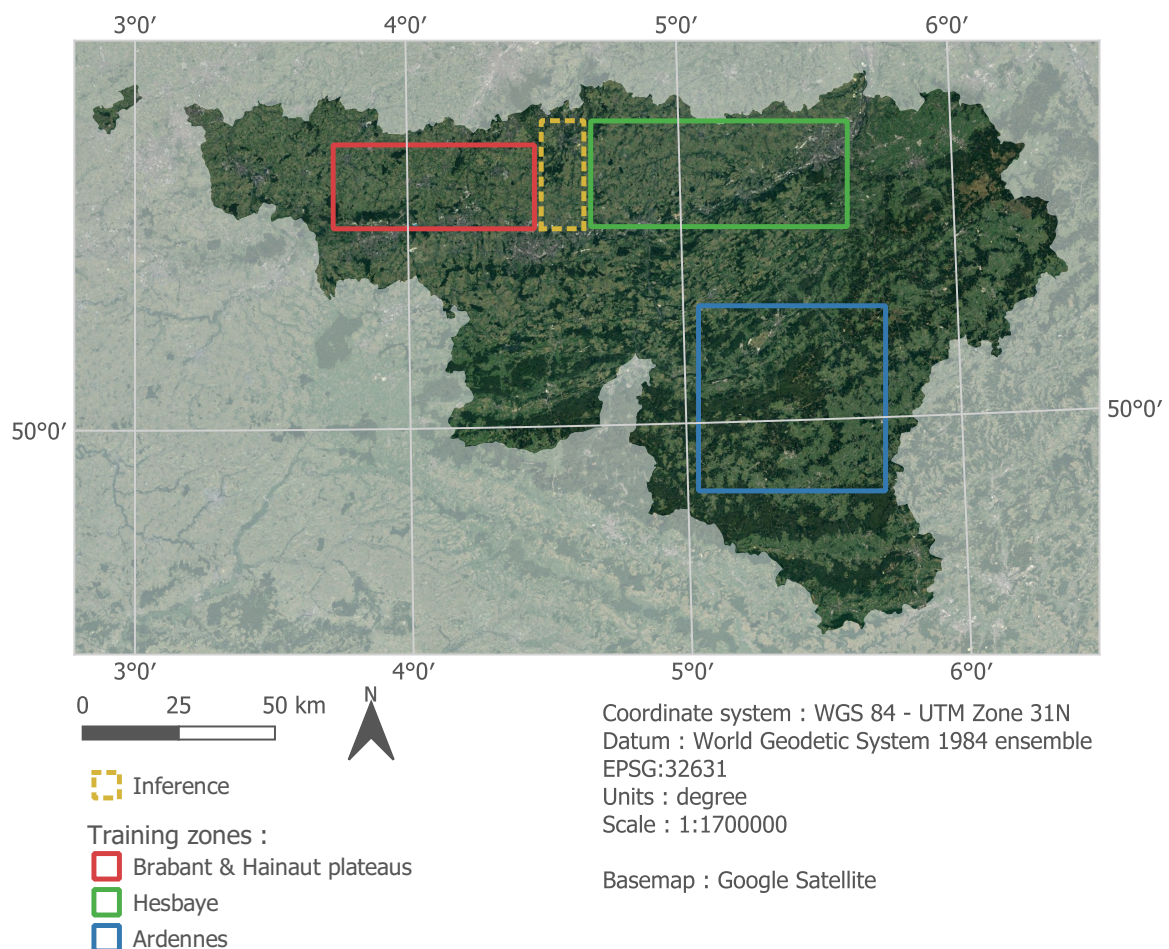


FIGURE 4.1: Location of the three training and inference zones forming the study area in Wallonia

4.2 Study period

The selected study period runs from September 2018 to September 2019, covering a full annual agricultural season. This duration is deliberately chosen to capture the diversity of crops, as the land is sown in both winter and summer. However, not all months of the year are included in the training of the ResUNet-a model for several reasons. Firstly, the optical imagery provided by Sentinel-2 is hindered by the clouds and other atmospheric phenomena. As a result, some monthly composites cannot be used due to the presence of significant cloud cover. The second reason is the computational cost of using too many images to train the model. Therefore, only certain months are selected, namely September, April, May, June and July. September is crucial to capture the soil when winter crops are sown, while April provides information on their growth and spring sowing. May, June and July correspond to the summer growing season.

It should be noted that the lack of images at certain times of the year is a legitimate criticism in the context of this study. In fact, certain seasons are completely absent from the table, in particular winter and spring. However, the aim of this work is only to evaluate whether the training of the ResUNet-a model can do without a certain percentage of field data for its training by replacing it with algorithmic data, and therefore does not aim to optimize an efficient ready-to-use model.

4.3 Data collection

4.3.1 Satellite data

The choice of satellite data is crucial in the evaluation of an artificial intelligence model and especially for the monitoring of agriculture which evolves over time and depending on the weather. For this reasons, it is necessary to choose a satellite with a high frequency revisit and a sufficient image resolution to discriminate the objects to identify. Sentinel-2 imagery is chosen. It has a five-day global revisit frequency that allows to obtain a higher probability of obtaining cloud-free images. In addition, Sentinel-2 images have a resolution of 10 meters for the red, green and blue bands as well as the near-infrared (NIR), which is important to be able to identify the smallest agricultural fields. All the Sentinel-2 bands are summarised in Table 4.1.

TABLE 4.1: Spectral bands of Sentinel-2 and its resolutions

Band	Band name	Wavelength range [μm]	Resolution [m]
B01	Coastal/ Aerosol	0.43 - 0.45	60
B02	Blue	0.45 - 0.51	10
B03	Green	0.53 - 0.59	10
B04	Red	0.64 - 0.68	10
B05	Vegetation Red Edge	0.70 - 0.74	20
B06	Vegetation Red Edge	0.74 - 0.78	20
B07	Vegetation Red Edge	0.78 - 0.85	20
B08	NIR (Near-Infrared)	0.85 - 0.88	10
B08a	Narrow NIR	0.85 - 0.88	20
B09	Water Vapor	0.93 - 0.96	60
B10	SWIR - Cirrus	1.36 - 1.38	60
B11	SWIR	1.57 - 1.66	20
B12	SWIR	2.10 - 2.29	20

The images from September 2018 to September 2019 are acquired by ESA and provided by the Geomatic research lab of UCLouvain. Only the Level-2A (L2A) images of the bands B02, B03, B04 and B08 are extracted. The L2A product provides orthorectified Bottom-Of-Atmosphere reflectance, with sub-pixel multispectral registration. It also includes the Scene Classification (SCL) masks, which are used to categorise or classify different types of land cover or clouds (see section 4.4.1).

4.3.2 LPIS

The LPIS for Wallonia, namely the *Système Intégré de Gestion et de Contrôle* (SIGeC), is managed by the Walloon public service (SPW), which includes the identification of farmers and the parcels they farm. This reference data layer is provided on an annual basis. The crops and plots declared in place on 31 May of the year are listed in a shapefile vector format, which is made available to the public in the course of the following year in the form of an anonymous parcel layer. The anonymous agricultural parcel layer used for the models' training is the SIGeC for the year 2019.

4.4 Preprocessing

Once the Sentinel-2 images are collected, a number of steps are required to obtain the inputs needed to train the ResUNet-a model and carry out the segmentation on eCognition. An overview of the preprocessing stages can be seen in Figure 4.2. The 5-layer

NDVI composite is used for ResUNet-a while the multiband composites are used for eCognition. All steps are performed using Python with the numpy and rasterio libraries. The bands selected for this study are the 10m resolution B02, B03, B04 and B08 and the SCL resampled from 20m to 10m.

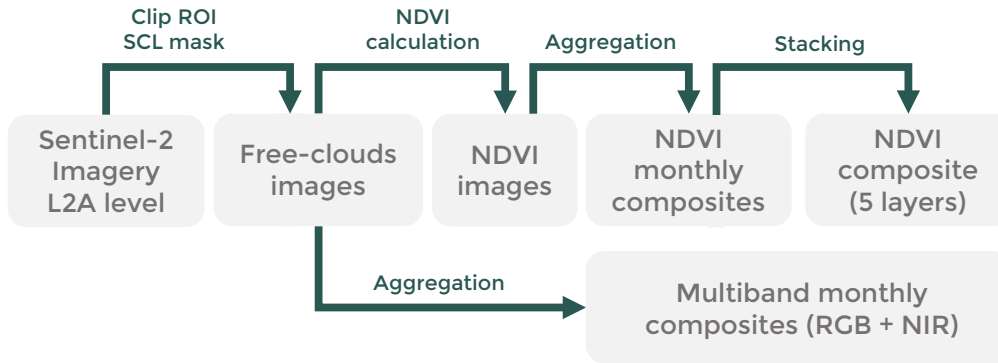


FIGURE 4.2: Overview of the preprocessing workflow

4.4.1 Cloud mask application

The first step is to clip the Sentinel-2 tiles according to the region of interest and then apply the SCL mask. The mask is generated using different spectral bands in the Sentinel-2 image and is used to identify different features on the Earth's surface, including the presence of clouds, cloud shadows, water, snow, etc. Each pixel in the image is assigned to a specific class based on its spectral content, only the vegetation, no vegetated, water and unclassified classes are retained in this work.

The red, blue, green and near-infrared bands have a resolution of 10 meters. Since the SCL mask has a resolution of 20 meters, the cloud masks must first be upsampled from 20 to 10 meters before being applied, using the nearest neighbour resampling method. The different images generated are saved in TIF format for each individual band.

4.4.2 Normalised Difference Vegetation Index calculation

The next step is to calculate the Normalised Difference Vegetation Index (NDVI) from measured reflectances in the red and near-infrared bands (B03 and B08 respectively). The NDVI is a widely used numerical indicator in remote sensing that quantifies the presence of healthy green vegetation in a given area, ranging from -1 to 1, with higher values indicating healthier vegetation. It is particularly useful for monitoring changes in vegetation over time and space.

The NDVI is calculated as follows:

$$NDVI = \frac{(NIR - Red)}{(NIR + Red)} = \frac{(B08 - B03)}{(B08 + B03)} \quad (4.1)$$

4.4.3 Composites

From the NDVI images generated, all images belonging to a given month of the year are combined to compute the values forming a monthly composite for the three training zones and the inference zone. The median method allow to combine the valid NDVI values and to exclude pixels classified as clouds. The median computation limits the impact of extreme values for clouds potentially unmasked, which would greatly affect the final result of the composite if the mean method were used. Each composite is assessed visually in QGIS, with quality assessment based on several criteria: overall cloud coverage and potential poorly cloud mask correction (presence of artefacts or unmasked clouds).

A monthly composite is only selected if the respective composites for all the zones for the same month meet the quality criteria. Therefore, the composites for the following months are selected for the four zones: September, April, May, June and July.

For each zone, the median monthly composites are stacked to obtain a single image of five layers (one layer for each month), resulting in four composites of five layers each. Based on the full-size composites of the three training zones (Figure 4.3), the images are also truncated to obtain composites with 2, 3, 4, 5, 10, 25, 50 and 75% of field data remaining in order to train the ResUNet-a model with different volume of field data (see section 4.6.1 for more explanations). Before the storage in TIF format, all the NaN values are converted to -1 in order to fit with the model.

In addition, multiband composites are generated from the cloud-free images, still using the median method, to perform the eCognition segmentation (see Section 4.5). Multiband composites include the RGB and NIR bands.



FIGURE 4.3: The 5-layer NDVI composites of the three training zones and their cut-outs

4.5 eCognition segmentation

4.5.1 Software

The algorithms used to segment and classify agricultural plots are based on *eCognition*. Marketed by Trimble, it's an advanced commercial software designed to automate the analysis and interpretation of satellite imagery. It provides a range of capabilities, including pixel-based object creation, production of larger objects from existing objects, image segmentation and object classification.

4.5.2 Field segmentation and classification

The input used for the segmentation is a monthly multiband composite of April 2019, which includes all three training zones. The choice of the image is based on a visual quality criterion, the April image is thus chosen because of its minimal cloud cover and the great local contrast between parcels it provides. The use of multiband imagery allows optimal use of the segmentation algorithm, as the weight of each image layer can be adjusted in its operations. Prior to segmentation, the NDVI is calculated from the red and infrared bands, enriching the image with layers of red, green, blue, near-infrared and NDVI.

MRS algorithm generates polygons on the image according to predefined parameters. Once these parameters have been defined, the algorithm groups the pixels or objects at each iteration to form clusters that best meet the specified criteria. The parameters include the shape, compactness and scale.

The segmentation is performed in several steps, as shown in Figure 4.4, and illustrated further in Figure 4.5. An initial segmentation is performed at a pixel scale to create pixel clusters in the form of small objects. A classification is then performed to distinguish between field and non-field areas, thus improving the second segmentation of the plots at an object scale.

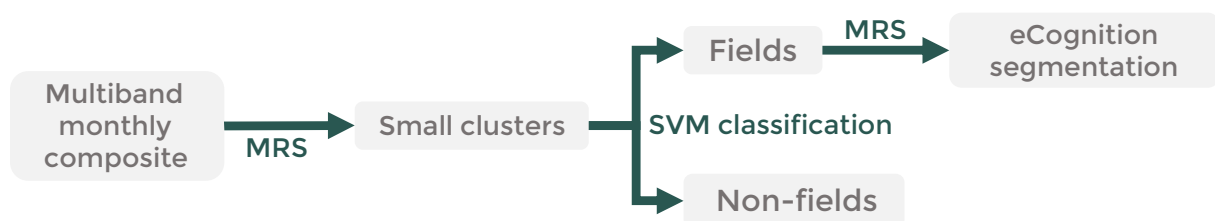


FIGURE 4.4: Overview of the eCognition segmentation

The classifier used is a SVM algorithm, chosen because of its ability to handle small training datasets, unlike most supervised training algorithms as explained in section 2.2.2. In the context of this work, it's important to be able to do a segmentation without a significant volume of training data, and therefore to be able to generate a small training dataset in a short time. To train the classifier, 15 field samples and 10 non-field samples are manually selected, all corresponding to clusters of the first segmentation and representing as many different patterns as possible. The classification is assessed by visual interpretation, distinguishing the majority of fields and non-fields without excessive optimisation, in order to avoid the exclusion of fields existing but not recognized by the algorithm.

A second segmentation is then performed at the object level by applying a filter to consider only the class of fields. This step involved grouping the clusters created during the first segmentation into larger objects, potentially representing cultivated parcels.

4.5.3 Parameters

The main parameters to be taken into account for a MRS are the scale, the shape and the compactness. The scale determines the interspectral variation between objects and therefore their size. A higher value results in more variation and therefore larger objects. For the shape, the value must be between 0 and 1. A value close to 0 favours colour as the determining factor in agglomeration, while a value close to 1 gives more weight to the shape homogeneity of the objects. Lastly, the compactness must be also between 0 and 1. A value close to 1 corresponds to a circular object, while a value close to 0 is suitable for more elongated shapes. In addition to these three main parameters, the weight allocated to each layer of the image can be modified, the higher the weight which is assigned to a layer the more of its information is used during the segmentation process. All the parameters used are shown below in Table 4.2.

TABLE 4.2: Parameters used for the MRS at the pixel and object levels

Parameters	Pixel level	Object level
Scale	20	40
Shape	0.6	0.7
Compactness	0.8	0.6
Layers	Weights	
Red	1	
Green	1	
Blue	1	
NIR	1	
NDVI	5	

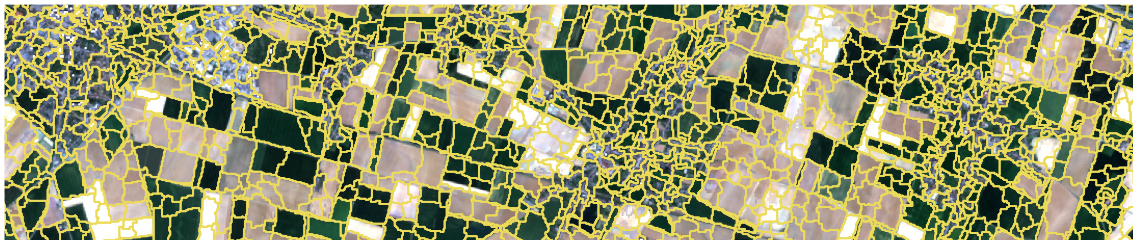
In eCognition, these parameters are chosen iteratively by trial and error, given that no universal method has been found to identify the most optimal segmentation parameters as explained in section 2.1.4. The final result is therefore chosen on the basis of an overall observation of all agricultural plots in the three training zones.

For classification, the SVM uses a linear kernel and a C parameter of 2, which is the default value proposed by eCognition. According to the software, the C parameter is a regularisation parameter that controls the trade-off between achieving a low training error and a low testing error, thus influencing the penalty for misclassifying training examples.

Orthophoto (multiband monthly composite of April 2019)



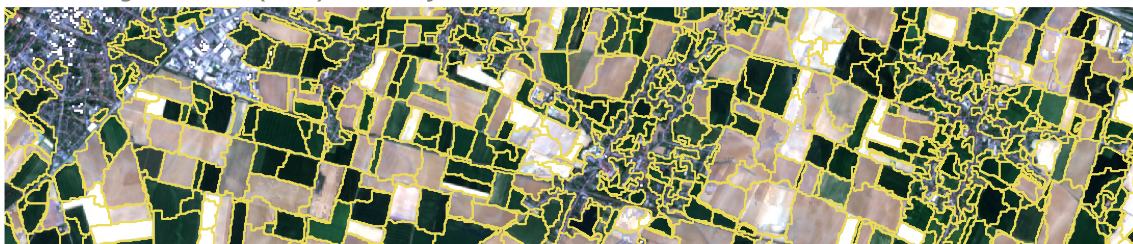
First segmentation (MRS) at a pixel level



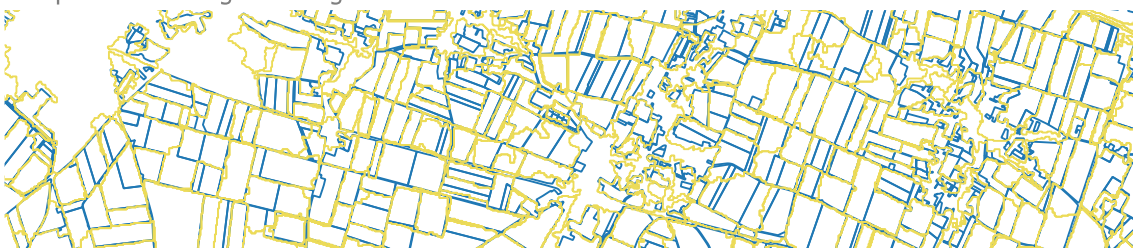
SVM classification (Fields vs. Non-fields)



Second segmentation (MRS) at an object level



Comparison of eCognition segmentation and LPIS 2019



— eCognition segmentation — LPIS 2019 — Fields — Non-fields

FIGURE 4.5: Illustration of the segmentation and classification stages performed on eCognition software

4.6 ResUNet-a

4.6.1 Method overview

To pursue the objective of this Master's thesis, the ResUNet-a model is used and trained several times independently with different input data for comparison purposes. For the sake of clarity, these different versions of the model are referred to as *projects* so as not to confuse them with the ResUNet-a model in its general sense.

The method consists of comparing the predictions of different projects, each trained with a different type of training and volume of field data. To make the following clear, two types of input are required to train a model: raster images corresponding in this thesis to the 5-layer NDVI composites, and a shapefile of the field plots (the eCognition segmentation or the LPIS, depending on the project). The Figure 4.6 illustrates the whole methodology for a better understanding.

Initially, the *Pre-training* project is built. This project consists of a self-supervised learning, i.e the model is trained on data generated by an algorithm (the eCognition segmentation). Once the Pre-training project is obtained, its learning is transferred to new projects called *Mix*. Based on the pre-training, these Mix projects are in turn trained with supervised learning, i.e on a set of input data where each field plot has an associated label thanks to the LPIS, corresponding to the field data. They are therefore trained with both self-supervised and supervised learning. The different Mix projects differ solely in the amount of field data used during their supervised training. These different projects are called Mix 100, Mix 75,... depending on the percentage of field data used as input for training the projects. To reduce the amount of field data, the composites are truncated as described in section 4.4.3.

A final set of projects is then constructed and trained with supervised learning only. These projects are trained from scratch, i.e. without transferring the learning from the Pre-training project. These projects are called Scratch 100, Scratch 75,... in the same way as the Mix projects. A Mix project and a Scratch project with the same percentage are thus trained on the same composites (Figure 4.3), but the first includes a pre-training stage with eCognition segmentation and the second does not.

Once trained, each project makes a prediction on an inference image, a 5-layer NDVI composite, corresponding to new data. The accuracy of these predictions is then evaluated using the metrics OA, F1-Score, MCC, IoU and $\text{RecRate}_{\text{soft}}$. All the intermediate stages are explained below.

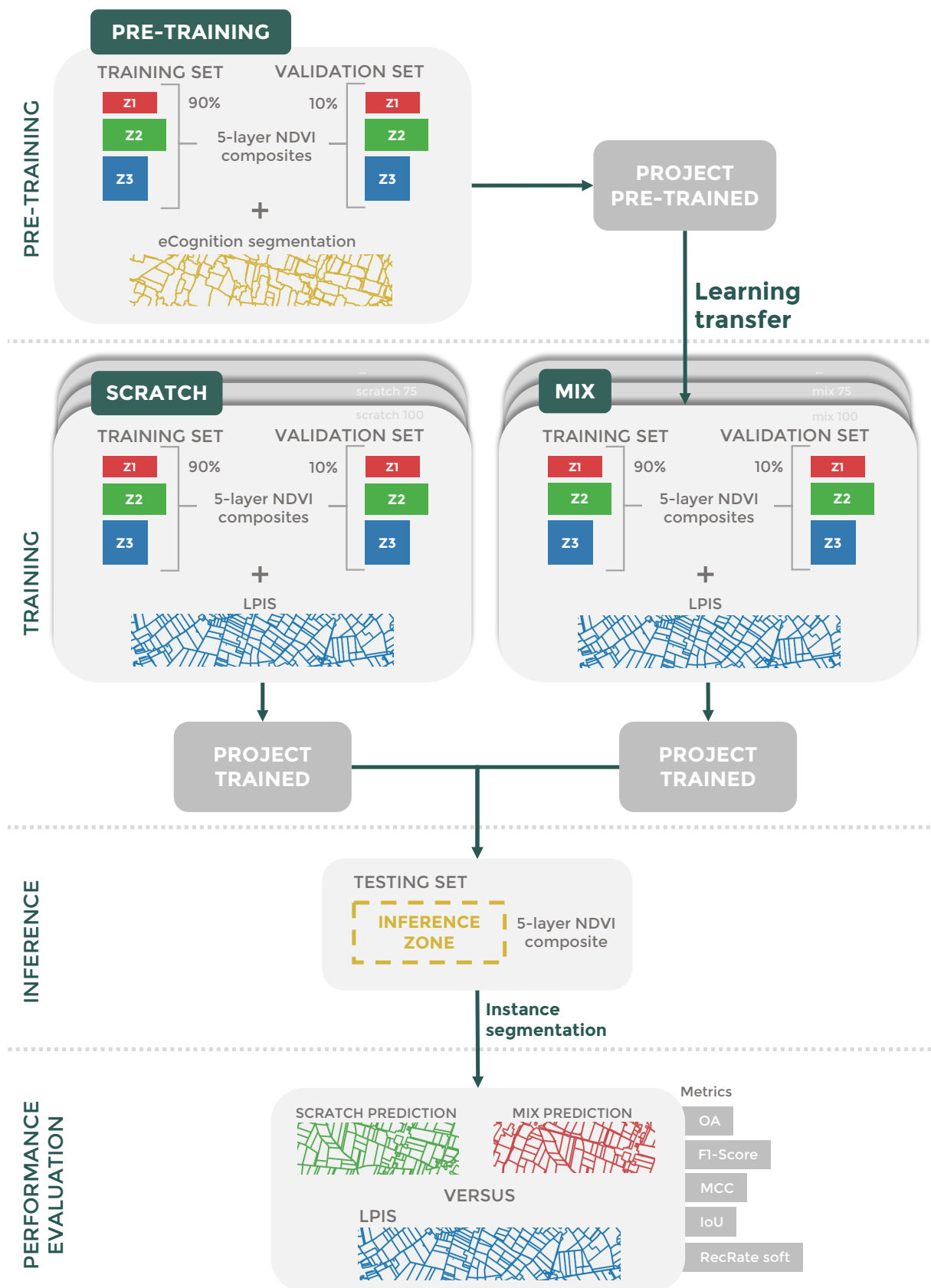


FIGURE 4.6: Overview of the method used from the training to the performance evaluation of ResUNet-a

The projects created can thus be summarized as follows :

Pre-training : this project involves training the model solely with algorithmic data, i.e. the eCognition segmentation. It serves as the basis for the creation of the Mix projects.

Mix 100, 75, 50, 25, 10, 5, 4, 3 and 2 : these projects include the transfer learning of the Pre-training project, on which it is then trained using respectively 100%, 75%, 50%, 25%, 10%, 5%, 4%, 3% and 2% of field data.

Scratch 100, 75, 50, 25, 10, 5, 4, 3 and 2 : these projects involve training the model solely with field data, i.e. the LPIS shapefile. The Scratch 100 can be considered as the baseline as it is trained only with field data and the full-size composites. The number of each project corresponds to the percentage of field data used as input.

The Figure 4.7 shows a visual summary of the inputs used in each project.

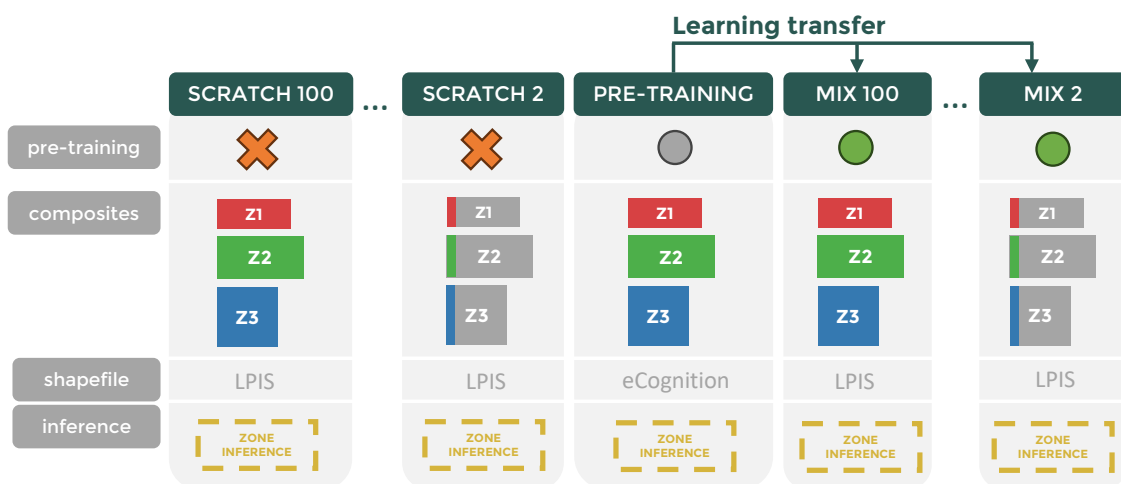


FIGURE 4.7: Summary of the inputs of each project

4.6.2 Model

The ResUNet-a model is built using the mxnet deep learning library under the GLUON API (Diakogiannis et al., 2020) and is used in the context of this thesis through the Center for High Performance Computing and Mass Storage (CISM) of UCLouvain. The model used in this thesis corresponds to that employed by Diakogiannis et al. (2020), the ResUNet-a d7 model.

There are 5 stages to run the model :

1. Creation of an instance mask from a shapefile
2. Generation of training data from the image and instance mask pair
3. Training of a ResUNet-a model
4. Inference with a trained model
5. Generation of fields in a vector format with instance segmentation

The five steps used for each project are described below.

4.6.3 Instance mask and training data creation

Based on reference shapefile and rasters which hold the targeted characteristics, an instance mask is created and paired with each small chip produced by chopping the raster images. The reference shapefile and the rasters used for the different projects differ and are summarized in the Figure 4.7.

Once satellite images and instance masks are paired, training and validation data can be extracted from them. The training set is used by definition to train the model while the validation set serves as a separate subset for fine-tuning the hyperparameters and providing an unbiased evaluation of the model during training by avoiding overfitting. Images are automatically split into validation (10%) and training sets (90%). Image areas with missing values are automatically not used as training or validation data.

The version used is ResUNet-a in multitasking mode, that is, the model learns multiple tasks at once. The model thus produces four correlated tasks, as illustrated in Figure 4.8 :

1. The field extent mask (the probability of each pixel to be in a field)
2. The field boundary mask (the probability of each pixel to be a boundary),

3. The distance to the closest boundary
4. The reconstruction of the input image in the Hue-Saturation-Value space.

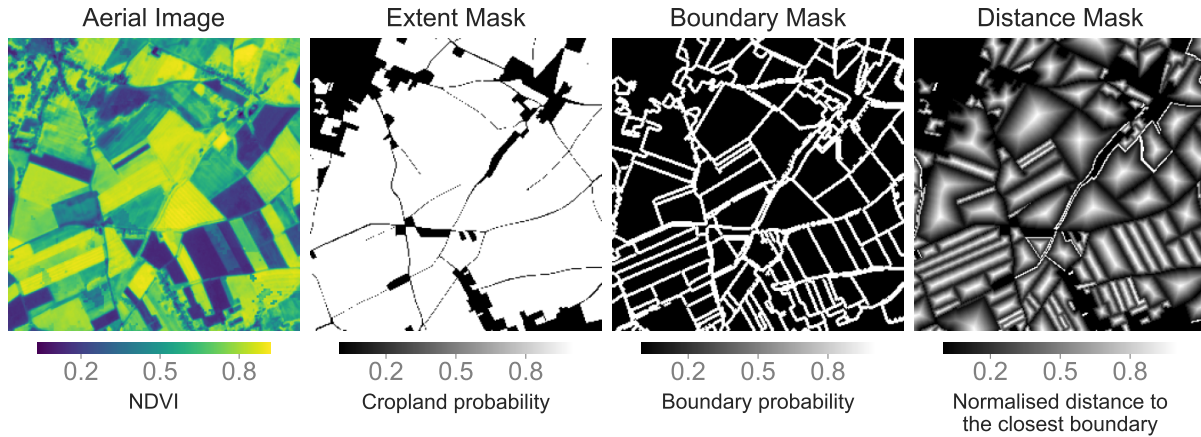


FIGURE 4.8: Example of a 256x256 chip with its related extent, boundary and distance masks

4.6.4 Training

Once the 256x256 chips have been created, a normalisation process is applied to the input images, as neural networks like to have inputs between 0 and 1. This helps the neural networks to converge faster during training and thus achieve better performance (Waldner and Diakogiannis, 2020). By default, the model use the training images to derive the means and standard deviations to use.

This step also includes a data augmentation processing to introduce variance in the training. As explained by Waldner and Diakogiannis (2020), this augmentation involves geometric transformations to ensure that the algorithm is exposed to slightly different images in each iteration. The purpose of data augmentation is to mitigate overfitting, a phenomenon in which a machine learning model adapts excessively to the training dataset, capturing specific details rather than effectively generalising and making predictions on images it has never encountered. During this augmentation process, each image and its corresponding ground truth mask are randomly rotated at different angles with a random centre, and zoomed in or out based on a random scale factor. Any parts of the image remaining outside the frame after these transformations are filled using reflect padding.

The projects are trained with an Adam optimizer (Kingma and Ba, 2017) and an initial learning rate of 0.001. Every 25 epochs, the learning rate decreases by a factor of

0.316. The projects are trained during 110 epochs and are optimised on the basis of the MCC metric on the extent of the 10% of data used for validation. The weights with the highest MCC on field extent within the 110 epochs is retained for each project.

4.6.5 Inference

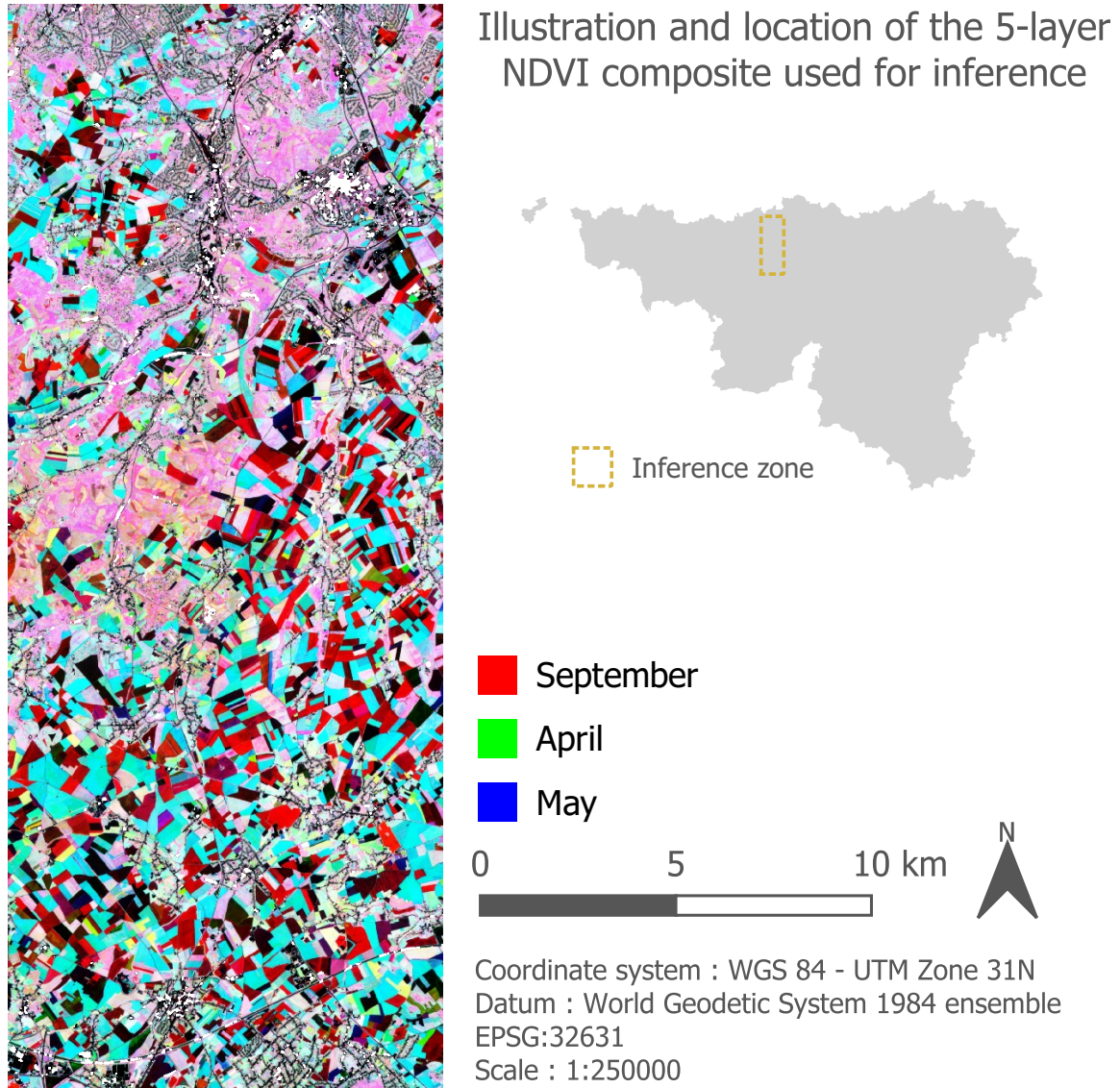


FIGURE 4.9: Illustration and location of the 5-layer NDVI composite used for inference

Once the training process is complete, the trained model can be used to make predictions on an inference image through a simple forward pass. The inference image of this study is a 5-layer NDVI composite that hasn't been used during the training (Figure 4.9).

Convolutional neural networks use contextual information to make predictions, and it is expected that their accuracy in classifying objects will vary depending on the location of the object within the input image. Objects located in the centre of the input images are expected to be well classified because their context is described uniformly in all directions. However, objects located near the edges of the input images may face challenges because they are missing parts of their context, potentially leading to misclassifications.

To address this issue, Diakogiannis et al. (2020) generate input images using a moving window of size 256 with a step size of 64 in each direction. This effectively varies the position of the image objects within the input images. The inference is then performed on all sets of input images and the results are averaged using the arithmetic mean. This approach aims to mitigate the potential misclassifications caused by the varying contextual information near the edges of the input images.

4.6.6 Instance segmentation for individual fields extraction

While the ResUNet-a convolutional neural network is capable of strictly identifying field boundaries, it is not necessarily capable of generating closed boundaries. An important step in extracting individual fields is therefore the application of instance segmentation.

Two post-processing methods introduced by Waldner and Diakogiannis (2020) include one based on thresholding and the other on watershed segmentation. Both use multiple semantic segmentation results and are data-driven, allowing automatic optimisation with a small reference sample. The cutoff method generates individual fields by thresholding and combining the boundary mask and the extent mask, while the watershed method extracts fields by all segmentation masks in a watershed algorithm. Waldner and Diakogiannis (2020) conclude that both approaches produce similar results.

The watershed segmentation is chosen for this work because of a deeper understanding of the method provided by previous readings on the watershed algorithm, where it has shown satisfactory results, ensuring its reliability (Watkins and van Niekerk, 2019; Wang et al., 2022a). The basics of the algorithm have already been explained in the previous section 2.1.3.

4.6.7 Performance evaluation

The different projects are assessed using metrics at pixel and object levels. The metrics used include OA, F1-Score, MCC, IoU and $\text{RecRate}_{\text{soft}}$, which are explained in the previous section 2.4, and are applied to the instance segmentation. The pixel-based metrics used are based solely on the field extent and not the field boundaries.

The resolution of the Sentinel-2 bands used is 10 metres. Therefore, field plots smaller than 0.09 hectares are excluded from the calculations to limit the impact of image resolution on the performance assessment.

Chapter 5

Results

This chapter provides a comprehensive overview of the outcomes derived from the methodology described in Chapter 4. The first section focuses on the training of the different projects in short, followed by the evaluation of their performance. The performance evaluation includes qualitative observations on the inference results as well as the analysis of the various metrics selected beforehand. The Mix and Scratch projects are first evaluated individually before their different performances are compared in order to highlight the effects of pre-training the ResUNet-a model. Finally, a comparison is made between the segmentation performed on eCognition and that performed by ResUNet-a.

5.1 Training

As mentioned above, the projects are trained during 110 epochs and optimised based on the MCC metric on the extent of the validation set. The weights with the highest MCC on the extent field within the 110 epochs are selected for each project, ensuring the selection of the most optimal model.

The Scratch 100 project, or baseline, is used as an example in Figure 5.1, which shows the different metrics that can be used to select model weights. It can be seen that the metrics give similar curves and would have had little impact on the choice of weights. The weights for the baseline project are selected at epoch 108 with a maximum MCC of 0.8513 for the field extent. Optimum training for all projects is usually achieved between the 100th and 110th epochs.

Figure 5.1 shows a number of significant downward peaks, most of which correspond to the start of a new learning rate cycle. A significant upward peak occurs from the fourth cycle onwards, after which variations in the MCC between epochs become marginal and where the optimal model is selected.

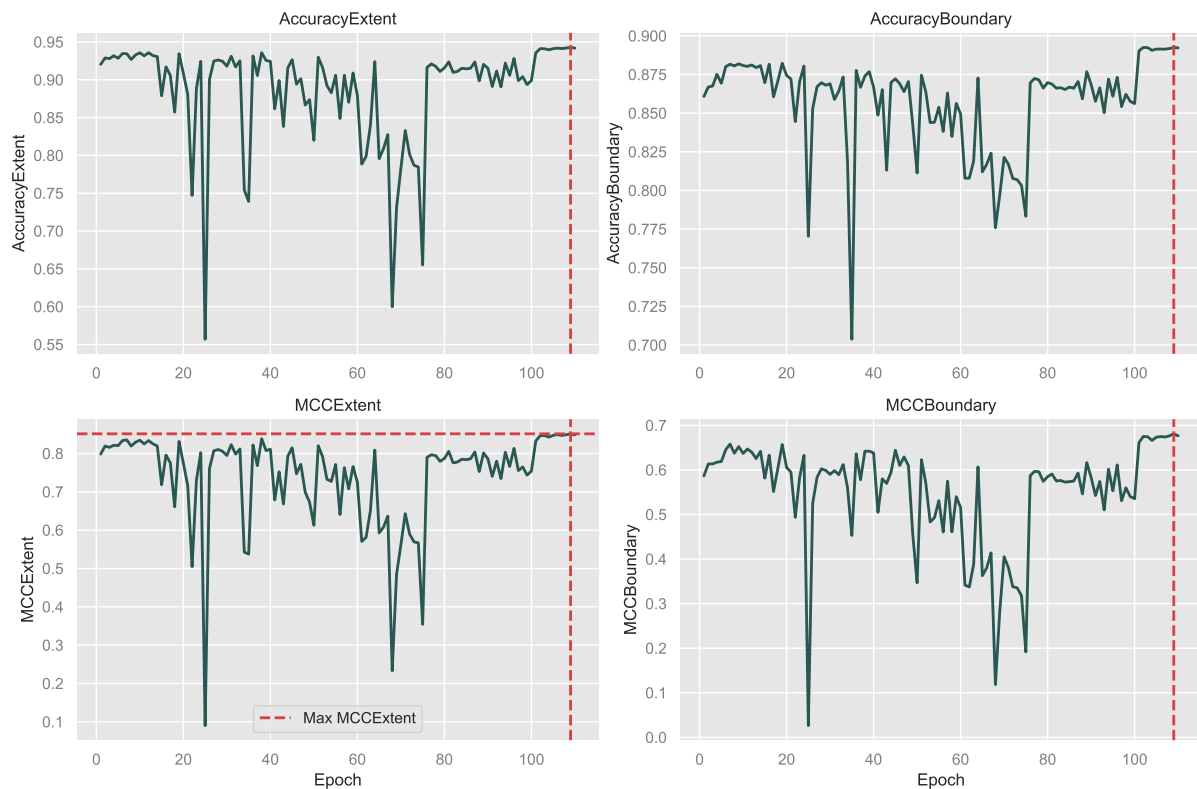


FIGURE 5.1: Evolution of the different metrics during the 110 epochs of training where the red line indicates the optimal model according to the MCC Extent

By cutting the 5-layer NDVI composites used as inputs for training the ResUNet-a model as explained in section 4.4.3, the number of labelled field plots could be reduced for training the different projects. Figure 5.1 summarises the number of labelled field plots used as inputs for each Mix and Scratch project, showing a decrease of over 100,000 labelled fields for training between projects containing 100% and 2% of the 5-layer NDVI composites.

TABLE 5.1: Number of field plots per zone used for training according to the project index, i.e. the percentage of field data used for training

Project index	Brabant & Hainaut plateaus	Hesbaye	Ardennes	TOTAL
100	30 460	37 050	39 746	107 256
75	23 799	31 282	27 985	83 066
50	16 924	22 550	18 293	57 767
25	8 179	11 503	10 082	29 764
10	3 936	5 786	3 736	13 458
5	2 056	2 846	1 995	6 897
4	1 658	2 376	1 723	5 757
3	1 311	1 814	1 440	4 565
2	922	1 175	1 088	3 185

5.2 Performance evaluation

As introduced in the previous sections, the different projects are trained on a training set corresponding to 90% of the three training zones with the remaining 10% serving as a validation set. The models are then evaluated on an inference zone corresponding to the test set, which comprises 2,522 field plots covering 311 km², i.e. approximately 3% of the study area in terms of number of fields and 6% in terms of surface area.

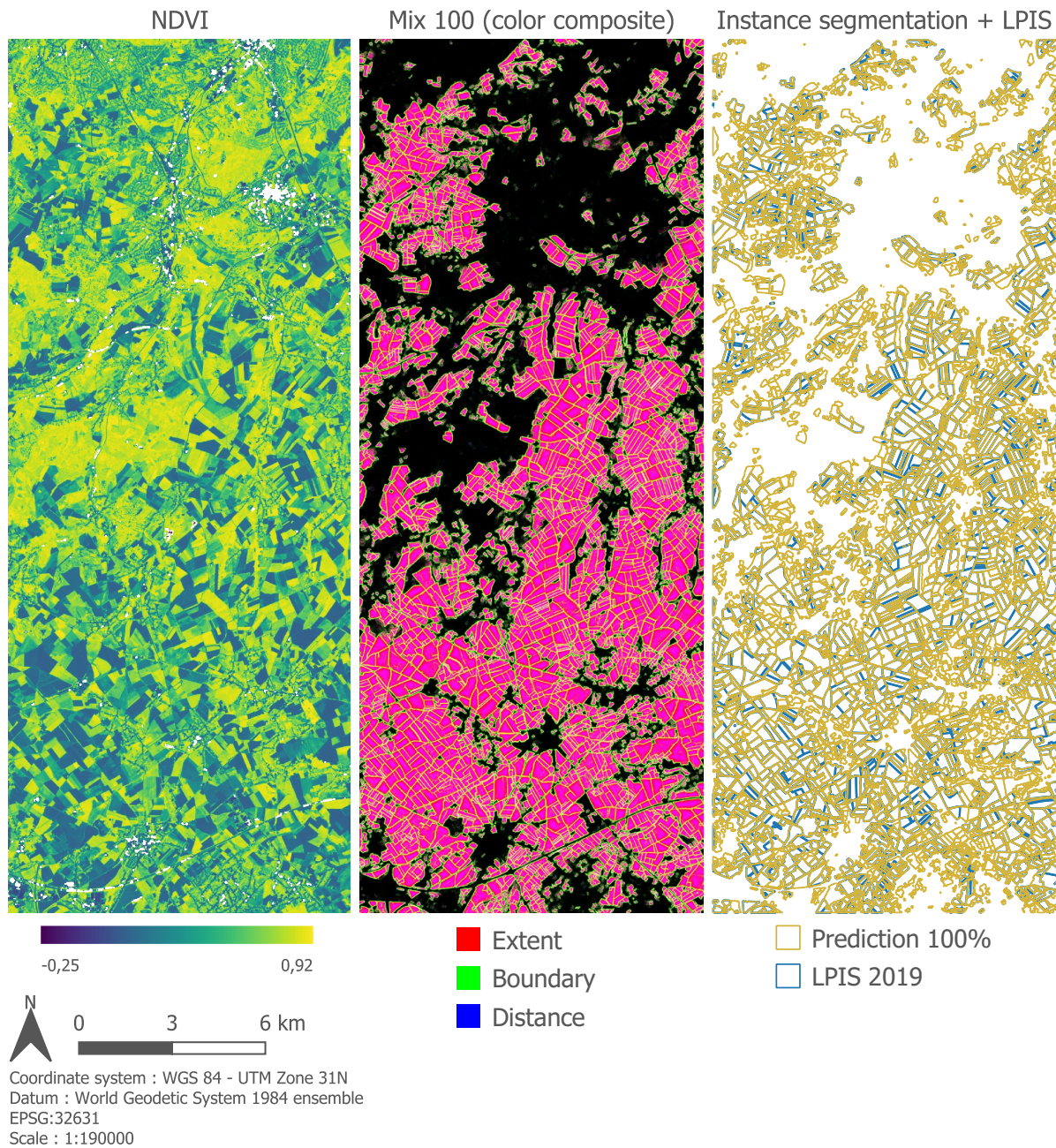


FIGURE 5.2: Example of results for the entire inference zone. From left to right: the NDVI monthly composite of April 2019, the prediction obtained for the Mix 100 project and the associated instance segmentation compared with LPIS 2019

An example of the results obtained for the entire inference zone is given above in Figure 5.2, but the interpretation is done on a zoom in Figure 5.3 for better visualisation.

Figure 5.3 shows an example of predictions where from the second to the fourth row of figures, the predictions are in the form of color composites representing the extent, boundaries and distance masks. At first glance, the predictions of the Mix and Scratch projects show only slight differences depending on the amount of field data used for training. Comparing the third and fourth rows of Figure 5.3, we can see that training with 2% of field data makes the field boundaries slightly blurred, the intermediate percentages are thus not shown in the figure due to the small gradient of variation. Comparing the different instance segmentations in the fifth and sixth row gives a clearer picture of the differences; the projects trained with 2% of field data more occasionally appear to over-segment the field parcels compared to the LPIS and to the projects trained with 100% of field data. In addition, the ResUNet-a model sometimes detects field plots that do not exist when compared to the NDVI image of the first row, this phenomenon is visible in the lower right part of the instance segmentations.

When comparing the Mix and Scratch projects, only minimal differences are visible to the naked eye. When comparing the segmentation instances, Scratch 2 seems to suffer more from noise than Mix 2. However, no statement about the effect of pre-training can be made on the basis of qualitative observation, a quantitative analysis is necessary.

Finally, the most glaring observation that can be made qualitatively is that the LPIS segmentation is not a perfect reference in itself. In fact, the second line of the figure highlights existing parcels that were detected by the ResUNet-a model but not declared and therefore not included in the LPIS. However, the number of undeclared parcels in the study area is not sufficient to have a real impact on model learning.

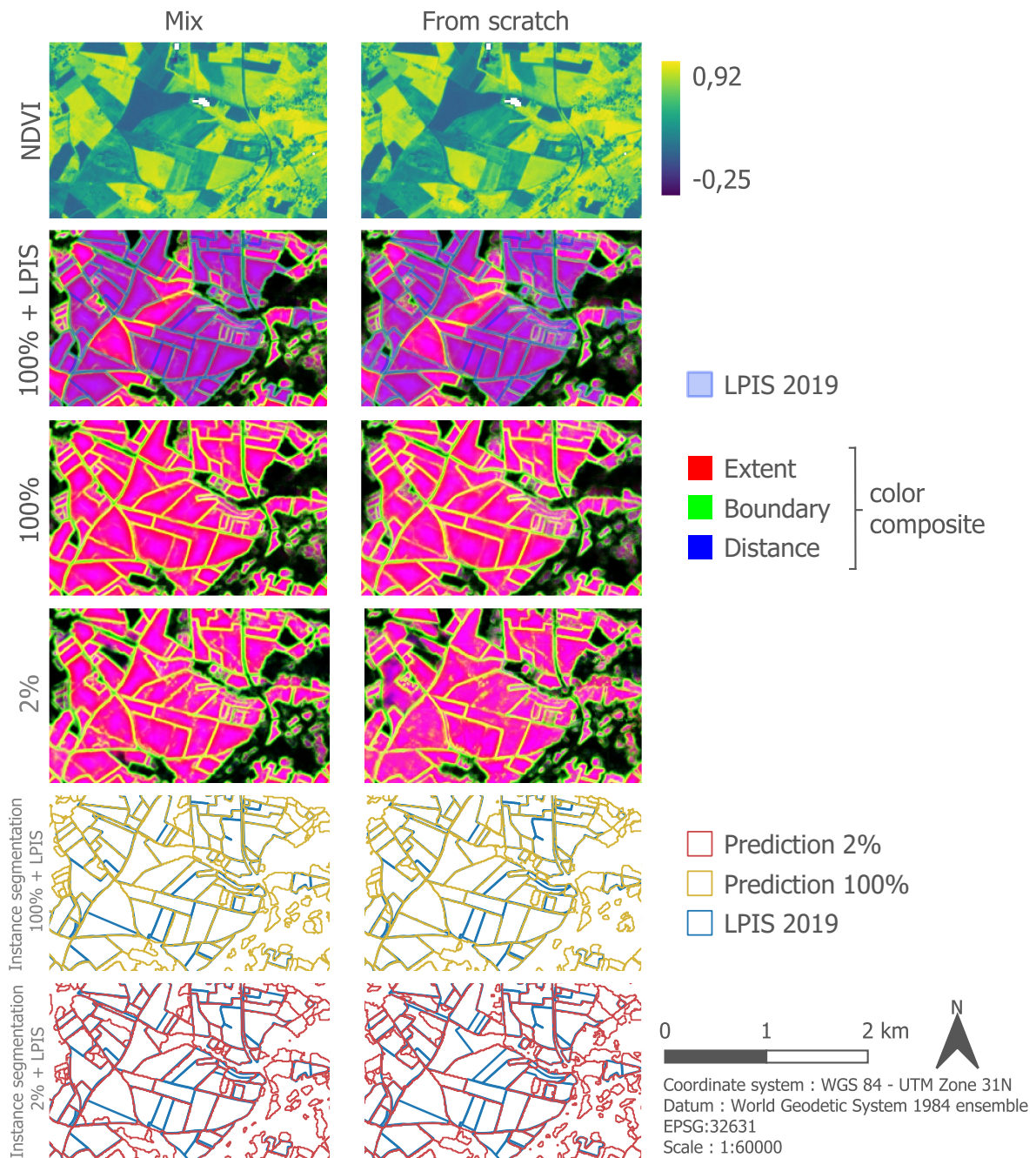


FIGURE 5.3: Zoom on the NDVI monthly composite of April 2019 and the predictions obtained for Mix and Scratch projects trained with 2% and 100% of field data, as well as the associated instance segmentations compared with LPIS 2019

The following sections provide a quantitative analysis.

5.2.1 Scratch projects

As shown in Table 5.2, the pixel-based metrics and the $RecRate_{soft}$ scores show very similar results regardless of the number of field plots used to train the projects. The standard deviation shows that the deviation from the mean is particularly low for OA and $RecRate_{soft}$. Furthermore, although this standard deviation is slightly higher for MCC and $DICE$, the analysis of the results by project shows no correlation between the amount of field data used for training and the performance of the ResUNet-a model. Nevertheless, both IoU_{50} and IoU_{median} have a higher standard deviation than the others. For the latter, we can see that the results decrease by about 14% and 12% respectively between the Scratch 100 and Scratch 2 projects, showing a decrease in model performance as the number of fields used for training decreases.

TABLE 5.2: Performance assessment on the pixel- and object-based metrics for the projects from scratch, i.e. without pre-training

Metrics	Scratch projects									mean	std dev
	100%	75%	50%	25%	10%	5%	4%	3%	2%		
OA	83.03	82.01	81.58	82.03	82.68	81.73	81.00	81.97	81.14	81.91	0.66
$DICE$	68.44	67.51	66.92	67.38	68.28	66.30	64.72	67.29	65.12	66.88	1.29
MCC	62.60	61.86	61.13	61.58	62.69	59.69	57.29	61.45	57.95	60.69	1.96
$RecRate_{soft}$	97.18	97.50	97.15	97.03	96.91	96.00	94.89	97.34	96.27	96.70	0.84
IoU_{50}	65.74	66.57	65.66	63.64	64.63	56.62	56.42	56.15	52.22	60.85	5.43
IoU_{median}	76.72	78.23	76.37	75.42	76.45	69.19	68.64	69.90	65.23	72.91	4.66

The different projects from scratch achieved an average OA of 82%. Although this value is satisfactory in itself, it is not sufficient to judge the real performance of the model. Indeed, if we compare it with the $DICE$ and MCC scores, which have average results of 67% and 61% respectively, we quickly see the limitations of OA , which is particularly sensitive to unbalanced classes. For object-based metrics, $RecRate_{soft}$ averages 97%. This softer measure ignores errors due to over- or undersegmentation, showing that this is the main problem the model suffers from. As mentioned above, the accuracy of the IoU_{50} increases from around 52% to 66% between training with 2% of field data and training with 100% of field data, while the accuracy of the IoU_{median} increases from around 65% to 77%. If we relate these results to the number of fields used to train the model, we see that around 10,000 additional field labels are needed to obtain an improvement of around 10%. Beyond 10,000 additional field labels, the performance improvement is marginal.

5.2.2 Mix projects

As shown in Table 5.3, the different Mix projects achieve an average Overall Accuracy of 82%, while the *DICE* and *MCC* metrics are around 67% and 61% respectively. For the object-based metrics, *RecRate_{soft}* averages 97%, still showing that the main errors are due to over- and under-segmentation. Both *IoU₅₀* and *IoU_{median}* show a decrease in model performance with the reduction in field data used for training, but the progression is less pronounced than for the Scratch projects, this can be seen in a lower standard deviation. There is a difference of 7% and 5% between Mix 2 and Mix 100 projects for *IoU₅₀* and *IoU_{median}* respectively.

TABLE 5.3: Performance assessment on the pixel- and object-based metrics for the Mix projects, i.e. with pre-training

Mix projects											
Metrics	100%	75%	50%	25%	10%	5%	4%	3%	2%	mean	std dev
OA	81.50	81.95	82.09	81.80	82.54	81.53	81.71	81.74	81.68	81.84	0.32
DICE	66.62	67.18	67.30	67.02	68.11	66.57	66.70	67.15	66.90	67.06	0.47
MCC	60.59	61.24	61.33	61.07	62.50	60.44	60.52	61.42	60.95	61.12	0.63
RecRate _{soft}	97.38	97.07	97.42	97.18	97.46	96.75	97.11	97.42	97.07	97.21	0.24
IoU ₅₀	65.74	65.86	64.99	63.20	63.01	61.26	60.31	61.78	58.76	62.77	2.47
IoU _{median}	77.12	77.00	75.73	74.78	75.26	74.46	73.47	74.78	72.05	74.96	1.60

5.2.3 Pre-training

The results for the pre-training assessment of the model, carried out solely with the eCognition segmentation, are lower than for the other projects (see Table 5.4). The *OA* is around 70%, while the *DICE* and *MCC* are around 53% and 43% respectively. For the object-based metrics, *RecRate_{soft}* equals 92%, and both *IoU₅₀* and *IoU_{median}* are around 23% and 32% respectively. These results are expected given that the pre-training is self-supervised.

TABLE 5.4: Performance assessment on the pixel- and object-based metrics for the pre-training

Metrics	Pre-training
<i>OA</i>	69.78
<i>DICE</i>	53.26
<i>MCC</i>	43.05
<i>RecRate_{soft}</i>	92.27
<i>IoU₅₀</i>	21.85
<i>IoU_{median}</i>	32.28

5.2.4 Training from scratch vs. transfer learning

Regarding to the pixel-based metrics, we observe in Figure 5.4 that the performance of the different projects varies very little with the number of field labels used for training. But also, the performance is almost identical with and without pre-training. The variations obtained between the Mix and Scratch projects for the pixel-based metrics are too small to confirm that pre-training has a real effect on the performance of the model.

Looking at the object-based metrics, both IoU_{50} and IoU_{median} , we can see that below a certain number of labelled fields, the performance of the Scratch projects drops drastically while pre-training limits the drop in model performance. Indeed, starting from 13,000 to 3,000 labelled fields used for training, the IoU_{50} and IoU_{median} drop by 13% and 11% respectively for the projects trained from scratch, whereas they drop by only 4% and 3% respectively when using transfer learning. For smaller training set sizes, the use of transfer learning in conjunction with an image segmentation algorithm can therefore significantly improve model performance. Over 13,000 labelled fields, performances with pre-training and from scratch are in the same order of magnitude.

The differences in results between the pixel-based and object-based metrics show that little data is required to obtain a deep learning model capable of distinguishing between fields and non-fields, but that the instance segmentation of the different fields is less effective with a smaller amount of training data. The $RecRate_{soft}$ remains relatively stable for all projects, indicating that projects without pre-training and with little data produce more under- and over-segmentation errors, and therefore poorer instance segmentation.

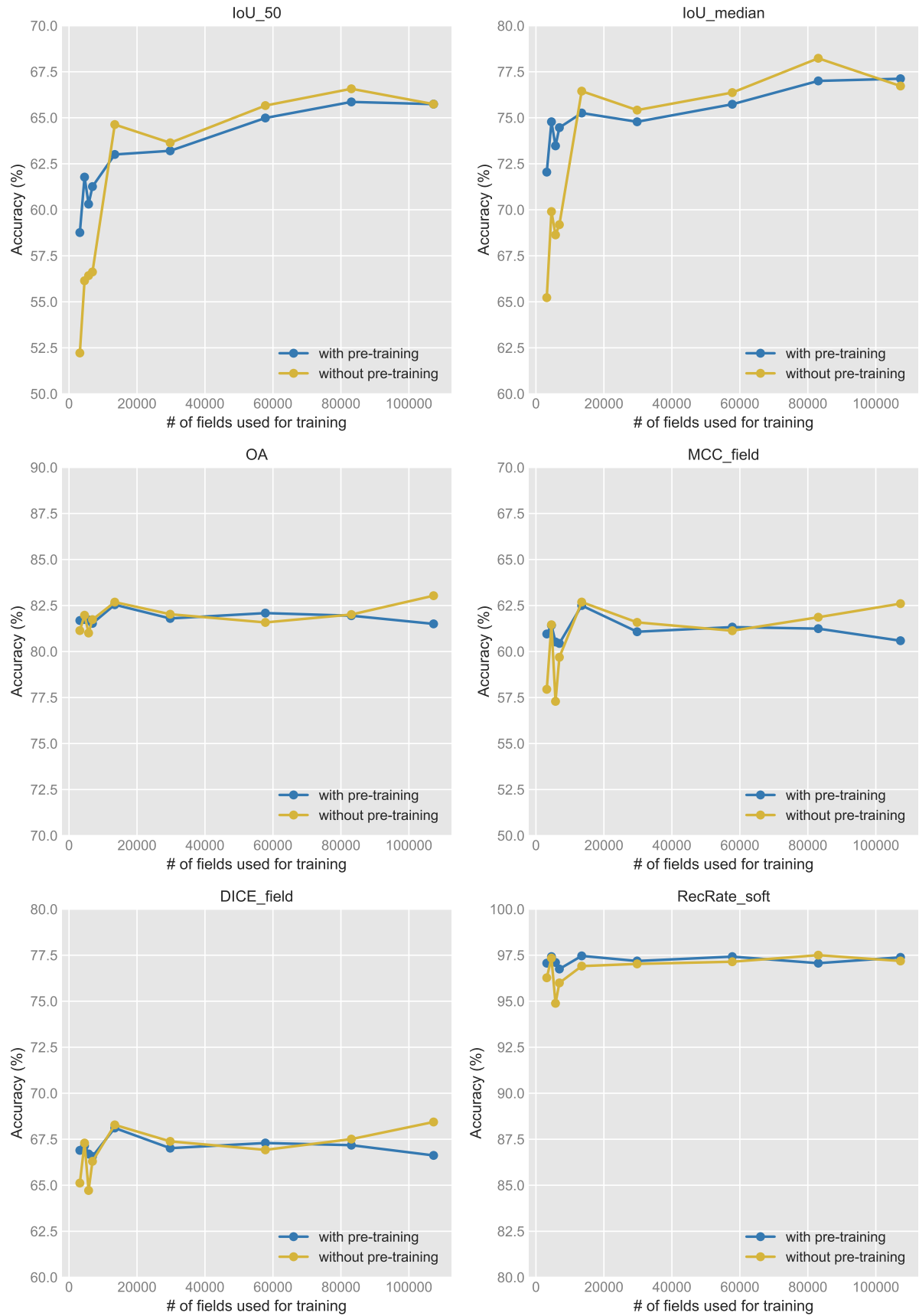


FIGURE 5.4: Evolution of pixel-based and object-based performance metrics according to the number of labelled fields used for training

5.2.5 eCognition vs. ResUNet-a

By applying the MRS algorithm to the inference area using the same method and the same parameters as for the training zones (see section 4.5), it can be seen visually in Figure 5.5 that the method cannot be generalised to other areas without refining the initial parameters. The result of the eCognition segmentation associated with the orthophoto shows the difficulty in determining the boundaries of the field parcels, giving rise to more random shapes than during the segmentation of the training zones (see Figure 4.5 in Chapter 4). There is also a poorer classification between fields and non-fields, as shown in the top right of the images in Figure 5.5, which has an impact on the final segmentation result.

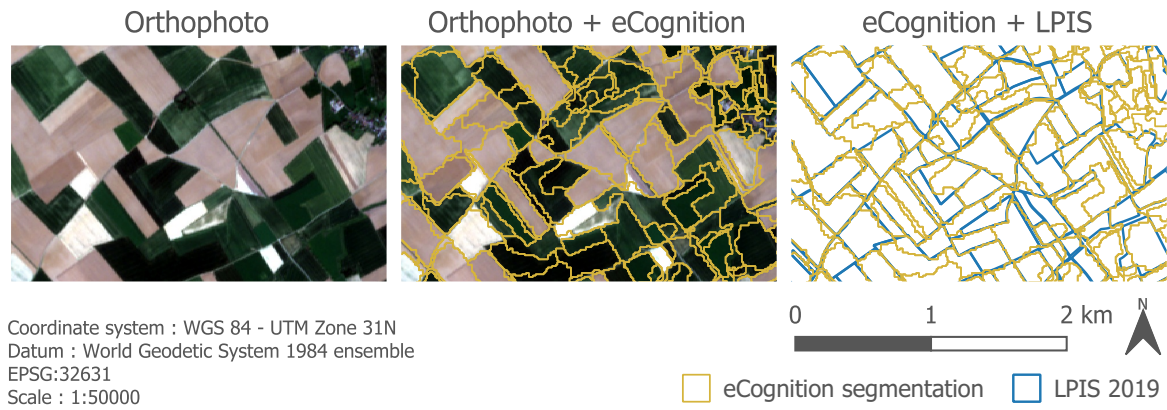


FIGURE 5.5: Illustration of the eCognition segmentation performed on the inference zone with the same method and parameters as the training zones

Quantitatively, the eCognition segmentation shows mixed results. For pixel-based metrics, the Overall Accuracy is around 59% and the *DICE* and *MCC* coefficients are around 48% and 38% respectively. For object-based metrics, the IoU_{50} is around 42% and the IoU_{median} is around 62%. The $RecRate_{soft}$, meanwhile, approaches 97%, as do the ResUNet-a projects.

TABLE 5.5: Comparison between eCognition, Pre-training, Scratch 2, and Mix 2 predictions

Metrics	eCognition	Pre-training	Scratch 2	Mix 2
<i>OA</i>	58.67	69.78	81.14	81.68
<i>DICE</i>	47.86	53.26	65.12	66.90
<i>MCC</i>	38.21	43.05	57.95	60.95
$RecRate_{soft}$	97.07	92.27	96.27	97.07
IoU_{50}	42.39	21.85	52.22	58.76
IoU_{median}	62.45	32.28	65.23	72.05

Comparing the segmentation performed on eCognition using the MRS algorithm and the pre-training project, which corresponds to training the ResUNet-a model only with the eCognition segmentation, we can see that the ResUNet-a pre-training performs slightly better for pixel-based metrics, but is much worse than the eCognition segmentation for object-based metrics. These results highlight the good ability of ResUNet-a to classify pixels into fields and non-fields, as well as its difficulty in accurately delineating field contours. However, if we compare the results of the eCognition segmentation with those of the Scratch 2 project, we can see that the ResUNet-a model, thanks to supervised training, quickly outperforms the classical segmentation algorithm, even on a small dataset (around 3,000 labelled fields in the case of the Scratch 2 project). As previously observed, the results of the Mix 2 project show that the transfer learning of a pre-trained model can further improve the performance of ResUNet-a on a small training dataset.

Chapter 6

Discussion

The results of the study provide a nuanced perspective on the feasibility and effectiveness of replacing field data with algorithmically generated data for training the ResUNet-a model. While some findings suggest potential challenges, others reveal promising insights and highlight the context-dependent nature of this approach.

6.1 Effect of training dataset reduction

Observations on pixel-based metrics reveal minimal disparity in model performance between the different projects trained, despite the reduction in the number of labelled fields used for training. Both Scratch and Mix projects show similar trends. While the Overall Accuracy shows promising results, the MCC and the DICE coefficient show more mixed results, which makes it difficult to draw any definitive conclusions.

Object-based metrics offer a more insightful analysis. Both IoU_{50} and $\text{IoU}_{\text{median}}$ scores decrease as the number of labelled fields used for training decreases. About 10,000 additional field labels are required to achieve an improvement of around ten percent, but the improvement becomes marginal beyond these 10,000 additional field labels. It suggests that the ResUNet-a model performs reasonably well even when the size of the training dataset is drastically reduced. The $\text{RecRate}_{\text{soft}}$, a softer measure that ignores the errors due to over- and under-segmentation, highlights some challenges for the model given its high scores. ResUNet-a performs well in semantic segmentation, effectively distinguishing between fields and non-fields, and thus demonstrating its ability in classification tasks, but has limitations in instance segmentation, where the individual delineation of fields is performed.

6.2 Effectiveness of pre-training

A comparison between the projects trained from scratch and the Mix projects underscore the contribution of pre-training to the performance of the model. As described above, a notable trend is the decline of IoU_{50} and IoU_{median} as the number of labeled fields for training decreases. This decline is more pronounced for projects trained from scratch compared to pre-trained projects. The biggest drop occurs between 13,000 and 3,000 labelled fields; While IoU_{50} and IoU_{median} experience a decrease of approximately 13% and 11% respectively for the projects trained from scratch, the decline is limited by 4% and 3% respectively for the pre-trained projects. Beyond 13,000 labeled fields, both pre-training and from scratch performances converge, suggesting that transfer learning has a substantial impact on smaller training sets but has a diminishing influence on larger datasets. This insight highlights the potential of transfer learning as a valuable tool, particularly in scenarios where data scarcity is a significant challenge.

6.3 Effectiveness of deep learning

Applying the MRS algorithm to the inference area, with the method and parameters used for the three training zones, reveals the main weakness of traditional image segmentation algorithms in the context of geographic object-based image analysis. The main drawback is the lack of a universal method for automating the selection of optimal values for image segmentation parameters. The use of subjective trial and error methods makes it difficult to develop a method that can be generalised to an entire large study area. The larger the study area, the more the field and landscape characteristics vary, making the selection of the few image segmentation parameters a complex task.

Deep learning offers an effective solution to this drawback, as it is able to automatically learn the relevant features from the input data, eliminating the need for manual feature refinement. This is particularly beneficial in scenarios where field boundary features can be complex and difficult to express using hand-crafted features, such as for large study areas. Deep learning also provides end-to-end learning, which eliminates the need for breaking down the task into multiple stages and designing separate algorithms for each step as was required for eCognition segmentation.

The results obtained by the ResUNet-a model are more promising, demonstrating the model's ability to identify field plots even with a small training dataset. Nonetheless, despite the limitations of traditional image segmentation algorithms, the results of this

thesis have shown that they can be useful to improve the performance of a deep learning model in the case of a lack of field data.

6.4 Limitations and error sources

The study carried out reveals certain limitations, suggesting possible improvements to increase the reliability of the results in the future.

The ResUNet-a model was pre-trained with algorithmically generated data using multi-resolution segmentation, and the SVM classifier was used for field and non-field classification. However, the choice of these algorithms was partly subjective, taking into account the literature review and the practicality of the different techniques. Although MRS and SVM algorithms are relatively easy to learn, other approaches have been shown to be effective. For example, Watkins and van Niekerk (2019) reports promising results from combining watershed segmentation with the Canny method, an edge detection operator. Selecting the best performing method would therefore require an in-depth study, including variations in the use of different algorithms as well as input images. A method may be effective on multiband composites, but less effective on NDVI, false colour or single-date images. In addition, the performance of algorithms can vary depending on the characteristics of the region studied. Improving field segmentation using algorithms requires a balance to be struck between the cost involved and the results obtained.

Another limitation stems from the specific methodology of ResUNet-a. Each project was trained only once, whereas it would have been preferable to subject them to several training cycles and to average their results in order to reduce their slightly random nature. However, this did not prevent the trends observed for each project from being highlighted.

In addition, other more general improvements can be considered, such as the use of higher resolution satellite imagery for more accurate detection of small field plots and narrow shapes. The optimisation of a more efficient cloud mask would also be beneficial to reduce the effects of artefacts and minimise the impact of unmasked clouds on image quality. In addition, it is always worth pointing out that the LPIS segmentation has some anomalies due to undeclared parcels, and that an error-free reference shapefile is preferable for an accurate assessment of the model's performance. However, there are not enough undeclared parcels in the study area to have a significant impact on the results of this work.

6.5 Perspectives

Future work should focus on improving the proposed approach by addressing the inherent limitations mentioned above and exploring broader applications. A critical area to be explored is the assessment of the transferability of the model to different geographical regions, with a focus on developing countries where resource constraints and limited data availability are common challenges. An iterative approach could also be developed, combining manual digitisation of agricultural parcels and pre-training, and iteratively enriching the training dataset based on the quality of the predictions obtained.

In conclusion, despite the existing challenges, the proposed pre-training approach is promising, especially when data limitations hinder deep learning capabilities. Its potential usefulness in developing countries, where resource constraints and data scarcity are prevalent, underlines the value of the results obtained. However, the contribution of pre-training using an algorithmically generated training dataset is strictly limited to regions where field data is almost non-existent.

More globally, the results of the study show the significant benefits of using a deep learning model to delineate agricultural parcels, not only for developing countries, but also for all countries that do not have an annual parcel system like the LPIS type. Using the ResUNet-a model significantly reduces the need for manual digitisation based on visual interpretation by up to 90% (from 100,000 to 10,000 labelled fields approximately). Manual digitisation is a costly process that requires the use of very high resolution data, such as VHR imagery, to ensure optimal accuracy. The deep learning approach therefore results in significant savings in both manpower and image acquisition costs, without compromising performance, which appears to be only slightly lower.

Despite the promising results, the transfer learning from a model trained in a region with abundant field data to regions with insufficient data, as studied by Wang et al. (2022a), could lead to better results than using traditional segmentation algorithms for pre-training. Certainly, an in-depth study of deep learning could contribute to the creation of appropriate and effective solutions that can improve agricultural monitoring practices and promote sustainable agricultural development in countries without annual parcel system.

Chapter 7

Conclusion

Accurate field delineation is critical to the implementation of up-to-date and accurate agricultural monitoring systems. Faced with challenges such as limited access to field data and modern technology, developing countries need innovative solutions to ensure effective monitoring of their agricultural systems. The traditional method of manual parcel delimitation, as used in Europe, is proving too costly, time-consuming and error-prone to be applicable in developing countries. In this context, the use of deep learning could meet the needs of these countries, but one drawback is that it requires a very significant volume of field data to be effective.

The main objective of this thesis was therefore to train a deep learning model using a classical unsupervised image segmentation algorithm, with the aim of minimising the need for large amounts of field data for training through the use of transfer learning. This research was carried out in Wallonia, Belgium, taking advantage of the large amount of field data provided by its LPIS.

Sentinel-2 images were collected and processed for the study. They were used to train the ResUNet-a model and to obtain a segmentation of agricultural parcels using a MRS and a SVM classification, both available in the eCognition software. A first ResUNet-a project was trained with the eCognition segmentation. Its learning was then transferred to other ResUNet-a projects, making them pre-trained and on which they were then trained with field data. Several comparisons were made, including between the performance of pre-trained and non-pre-trained projects, between different projects using different sizes of labelled field datasets, and between the classical image segmentation algorithm and the ResUNet-a model.

The results of this study provided a balanced perspective on the feasibility and effectiveness of the approach. In its general form, ResUNet-a performs well in semantic segmentation, effectively distinguishing between fields and non-fields, and thus

demonstrating its ability in classification tasks, but has limitations in instance segmentation, where the individual delineation of fields is performed. However, despite the reduction in the number of fields labelled for training, the ResUNet-a model proved surprisingly resilient, only registering a significant drop in performance when the input data was drastically reduced. This observation was true for both pre-trained and non-pre-trained projects. However, an important key finding is that there was a more noticeable drop in performance for models trained from scratch compared to those pre-trained, especially for small training datasets, while pre-training becomes less relevant as dataset size increases. These results highlighted the potential of this approach in regions where field data is almost non-existent.

The approach studied could therefore be beneficial to all countries that do not have an annual parcel system, enabling significant savings in both manpower and image acquisition costs by reducing the need for manual digitisation based on visual interpretation by up to 90%, with only a slight decrease in model performance. Deep learning proved to be a robust solution, offering automatic feature learning and end-to-end learning capabilities, overcoming the drawbacks of traditional segmentation approaches that rely on manual feature refinement and multiple stages processing.

Nonetheless, despite the promising results of the approach studied, in-depth study of deep learning, including the transfer learning from a region with abundant field data to another where data is scarce, could help to create more appropriate and effective solutions for providing countries, without annual parcel system like the LPIS, with accessible decision-making tools in the field of agriculture.

Bibliography

- Aldalil, Mohammed (2021). "A New Method for Measuring Leveling Heights Using Computer Vision Techniques". In: *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1–6. DOI: 10.1109/ICCCEEE49695.2021.9429674.
- Amidi, Shervine and Afshine Amidi (2019). *Convolutional Neural Networks cheatsheet*. URL: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>.
- Baatz, M., U. Benz, S. Dehghani, M. Heynen, A. Hölzje, P. Hofmann, I. Lingenfelder, M. Mimler, M. Sohlbach, M. Weber, and G. Willhauck (2004). "eCognition Professional: User Guide 4". In.
- Baatz, M. and A. Schäpe (2000). "Multiresolution segmentation - an optimization approach for high quality multi-scale image segmentation". In: *J. Strobl, T. Blaschke, G. Griesebner (Eds.), Angewandte Geographische Informationsverarbeitung XII (pp. 12–23). Wichmann-Verlag, Heidelberg*.
- Belgiu, Mariana and Lucian Drăguț (2016). "Random forest in remote sensing: A review of applications and future directions". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 114, pp. 24–31. ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2016.01.011>.
- Benz, Ursula, Peter Hofmann, Gregor Willhauck, Iris Lingenfelder, and Markus Heynen (2004). "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 58, pp. 239–258. DOI: 10.1016/j.isprsjprs.2003.10.002.
- Beucher, Serge and F Meyer (1993). "Segmentation: The Watershed Transformation. Mathematical Morphology in Image Processing". In: *Optical Engineering* 34, pp. 433–481.
- Blaschke, Thomas (2010). "Object based image analysis for remote sensing". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 65, pp. 2–16. DOI: 10.1016/j.isprsjprs.2009.06.004.
- Brodrick, Philip G., Andrew B. Davies, and Gregory P. Asner (2019). "Uncovering Ecological Patterns with Convolutional Neural Networks". In: *Trends in Ecology Evolution* 34.8, pp. 734–745. ISSN: 0169-5347. DOI: <https://doi.org/10.1016/j.tree.2019.03.006>.

- Canedo, Daniel and António Neves (2019). "Facial Expression Recognition Using Computer Vision: A Systematic Review". In: *Applied Sciences* 9. DOI: 10.3390/app9214678.
- Chi, Mingmin, Rui Feng, and Lorenzo Bruzzone (2008). "Classification of hyperspectral remote-sensing data with primal SVM for small-sized training dataset problem". In: *Advances in Space Research* 41, pp. 1793–1799. DOI: 10.1016/j.asr.2008.02.012.
- Chicco, Davide and Giuseppe Jurman (2020). "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: *BMC Genomics* 21. DOI: 10.1186/s12864-019-6413-7.
- Cremers, Daniel, Mikael Rousson, and Rachid Deriche (2007). "A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape." In: *Int J Comput Vision* 72, 195–215. DOI: 10.1007/s11263-006-8711-1.
- Deng, Jun, Xiaojing Xuan, Weifeng Wang, Zhao Li, Hanwen Yao, and Zhiqiang Wang (2020). "A review of research on object detection based on deep learning". In: *Journal of Physics: Conference Series* 1684.1, p. 012028. DOI: 10.1088/1742-6596/1684/1/012028.
- Diakogiannis, Foivos I., François Waldner, Peter Caccetta, and Chen Wu (2020). "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162, pp. 94–114. DOI: 10.1016/j.isprsjprs.2020.01.013.
- Duro, Dennis, Steven Franklin, and Monique Dubé (2012). "A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery". In: *Remote Sensing of Environment* 118, 259–272. DOI: 10.1016/j.rse.2011.11.020.
- Géron, Aurélien (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Inc. ISBN: 9781492032649.
- He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2020). "Mask R-CNN". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2, pp. 386–397. DOI: 10.1109/TPAMI.2018.2844175.
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- Hoeser, Thorsten and Claudia Kuenzer (2020). "Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends". In: *Remote Sensing* 12.10. ISSN: 2072-4292. DOI: 10.3390/rs12101667.

- Huang, C., L. Davis, and J. Townshend (2002). "An assessment of support vector machines for land cover classification". In: *International Journal of Remote Sensing* 23. DOI: 10.1080/01431160110040323.
- Ioffe, Sergey and Christian Szegedy (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. arXiv: 1502.03167 [cs.LG].
- Jong, Maxwell, Kaiyu Guan, Sibow Wang, Yizhi Huang, and Bin Peng (2022). "Improving field boundary delineation in ResUNets via adversarial deep learning". In: *International Journal of Applied Earth Observation and Geoinformation* 112, p. 102877. DOI: 10.1016/j.jag.2022.102877.
- Juneja, Mamta and Parvinder Sandhu (2009). "Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain". In: *International Journal* 1, pp. 614–621. DOI: 10.7763/IJCTE.2009.V1.100.
- Kaganami, Hassana Grema and Zou Beiji (2009). "Region-Based Segmentation versus Edge Detection". In: *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1217–1221. DOI: 10.1109/IIH-MSP.2009.13.
- Karpathy, Andrej (2016). "CS231n: Convolutional Neural Networks for Visual Recognition". In: URL: cs231n.github.io/convolutional-networks/.
- Kelly, Maggi, Samuel Blanchard, Ellen Kersten, and Kevin Koy (2011). "Terrestrial Remotely Sensed Imagery in Support of Public Health: New Avenues of Research Using Object-Based Image Analysis". In: *Remote Sensing* 3, pp. 2321–2345. DOI: 10.3390/rs3112321.
- Kim, M., Marguerite Madden, and Timothy Warner (2008). "Estimation of optimal image object size for the segmentation of forest stands with multispectral IKONOS imagery". In: pp. 291–307. ISBN: 978-3-540-77057-2. DOI: 10.1007/978-3-540-77058-9_16.
- Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG].
- Krishna, N. Murali, Ramidi Yashwanth Reddy, Mallu Sai Chandra Reddy, Kasibhatla Phani Madhav, and Gaikwad Sudham (2021). "Object Detection and Tracking Using Yolo". In: *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1–7. DOI: 10.1109/ICIRCA51532.2021.9544598.
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2001). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Communications of the ACM* 60.6, pp. 4477–4479. DOI: 10.1145/3065386.
- Li, Zewen, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou (2022). "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects". In: *IEEE*

- Transactions on Neural Networks and Learning Systems* 33.12, pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.
- Liu, Desheng and Fan Xia (2010). “Assessing object-based classification: Advantages and limitations”. In: *Remote Sensing Letters* 1, pp. 187–194. DOI: 10.1080/01431161003743173.
- Ma, Lei, Manchun Li, Xiaoxue Ma, Liang Cheng, Peijun Du, and Yongxue Liu (2017). “A review of supervised object-based land-cover image classification”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 130, pp. 277–293. ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2017.06.001>.
- Mahdi, Fahad Parvez, Kota L. Motoki, and Syoji Kobashi (2020). “Optimization technique combined with deep learning method for teeth recognition in dental panoramic radiographs”. In: *Scientific Reports* 10.19261. DOI: 10.1038/s41598-020-75887-9.
- Mantero, P., G. Moser, and S.B. Serpico (2005). “Partially Supervised classification of remote sensing images through SVM-based probability density estimation”. In: *IEEE Transactions on Geoscience and Remote Sensing* 43.3, pp. 559–570. DOI: 10.1109/TGRS.2004.842022.
- Meyer, Fernand (2012). “The watershed concept and its use in segmentation : a brief history”. In: *CoRR abs/1202.0216*. DOI: <https://doi.org/10.48550/arXiv.1202.0216>.
- Mittal, H., A.C. Pandey, and M. et al. Saraswat (2022). “A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets”. In: *Multimed Tools Appl* 81, 35001–35026. DOI: <https://doi.org/10.1007/s11042-021-10594-9>.
- Mountrakis, Giorgos, Junho Im, and Caesar Ogole (2011). “Support vector machines in remote sensing: A review”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.3, pp. 247–259. ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2010.11.001>.
- Muñoz, X, J Freixenet, X Cufi, and J Martí (2003). “Strategies for image segmentation combining region and boundary information”. In: *Pattern Recognition Letters* 24.1, pp. 375–392. ISSN: 0167-8655. DOI: [https://doi.org/10.1016/S0167-8655\(02\)00262-3](https://doi.org/10.1016/S0167-8655(02)00262-3).
- Myint, Soe W., Patricia Gober, Anthony Brazel, Susanne Grossman-Clarke, and Qihao Weng (2011). “Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery”. In: *Remote Sensing of Environment* 115.5, pp. 1145–1161. ISSN: 0034-4257. DOI: <https://doi.org/10.1016/j.rse.2010.12.017>.
- Ng, Andrew (2018). *CS230 Deep Learning*. URL: <https://cs230.stanford.edu/>.
- Niemeyer, Joachim, Franz Rottensteiner, and Uwe Soergel (2014). “Contextual classification of lidar data and building object detection in urban areas”. In: *ISPRS Journal*

- of Photogrammetry and Remote Sensing* 87, pp. 152–165. ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2013.11.001>.
- Oruc, Murat and Aycan Marangoz (2004). “Comparison of pixel-based and object-oriented classification approaches using Landsat-7 ETM spectral bands”. In: *Proc. ISPRS Conference, Istanbul, Turkey*, pp. 1–5.
- Padilla, Rafael, Sergio L. Netto, and Eduardo A. B. da Silva (2020). “A Survey on Performance Metrics for Object-Detection Algorithms”. In: pp. 237–242. DOI: 10.1109/IWSSIP48289.2020.9145130.
- Pal, Mahesh (2005). “Random forest classifier for remote sensing classification”. In: *International Journal of Remote Sensing - INT J REMOTE SENS* 26, pp. 217–222. DOI: 10.1080/01431160412331269698.
- Pavlova, Maria, Valerii Timofeev, Dmitry Bocharov, Irina Kunina, Anna Smagina, and Mikhail Zagarev (2021). “Segmentation of Agricultural Parcels in Satellite Images Based on Historical Vegetation Index Data”. In: *Preprints*. DOI: 10.20944/preprints202112.0243.v1.
- Persello, C., V.A. Tolpekin, J.R. Bergado, and R.A. de By (2019). “Delineation of agricultural fields in smallholder farms from satellite images using fully convolutional networks and combinatorial grouping”. In: *Remote Sensing of Environment* 231, p. 111253. ISSN: 0034-4257. DOI: <https://doi.org/10.1016/j.rse.2019.111253>.
- Roerdink, Jos and A. Meijster (2003). “The Watershed Transform: Definitions, Algorithms and Parallelization Strategies”. In: *Fundam Inf* 41. DOI: 10.3233/FI-2000-411207.
- Rong, Weibin, Zhanjing Li, Wei Zhang, and Lining Sun (2014). “An improved Canny edge detection algorithm”. In: *2014 IEEE International Conference on Mechatronics and Automation*, pp. 577–582. DOI: 10.1109/ICMA.2014.6885761.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv: 1505.04597 [cs.CV].
- Sheykhmousa, Mohammadreza, Masoud Mahdianpari, Hamid Ghanbari, Fariba Mohammadimanesh, Pedram Ghamisi, and Saeid Homayouni (2020). “Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13, pp. 6308–6325. DOI: 10.1109/JSTARS.2020.3026724.
- Shrivakshan, G.T. and Chandramouli Chandrasekar (2012). “A Comparison of various Edge Detection Techniques used in Image Processing”. In: *International Journal of Computer Science Issues* 9, pp. 269–276.
- Theodoridis, Sergios (2020). *Machine Learning: A Bayesian and Optimization Perspective, 2nd edition*. Academic Press. ISBN: 9780128188033.

- Vapnik, Vladimir (1998). "The Support Vector Method of Function Estimation". In: *Nonlinear Modeling: Advanced Black-Box Techniques*. Ed. by Johan A. K. Suykens and Joos Vandewalle. Boston, MA: Springer US, pp. 55–85. ISBN: 978-1-4615-5703-6. DOI: 10.1007/978-1-4615-5703-6_3.
- Waldner, François and Foivos I. Diakogiannis (2020). "Deep learning on edge: extracting field boundaries from satellite images with a convolutional neural network". In: *CoRR abs/1910.12023*. DOI: <https://doi.org/10.48550/arXiv.1910.12023>.
- Waldner, François, Foivos I. Diakogiannis, Kathryn Batchelor, Michael Ciccotosto-Camp, Elizabeth Cooper-Williams, Chris Herrmann, Gonzalo Mata, and Andrew Toovey (2021). "Detect, Consolidate, Delineate: Scalable Mapping of Field Boundaries Using Satellite Images". In: *Remote Sensing* 13.11. ISSN: 2072-4292. DOI: 10.3390/rs13112197.
- Wang, Sherrie, François Waldner, and David B. Lobell (2022a). "Unlocking large-scale crop field delineation in smallholder farming systems with transfer learning and weak supervision". In: *CoRR abs/2201.04771*. DOI: <https://doi.org/10.48550/arXiv.2201.04771>.
- Wang, Yuanbo, Unaiza Ahsan, Hanyan Li, and Matthew Hagen (2022b). "A Comprehensive Review of Modern Object Segmentation Approaches". In: *Foundations and Trends® in Computer Graphics and Vision* 13.2–3, 111–283. ISSN: 1572-2759. DOI: 10.1561/06000000097.
- Watkins, Barry and Adriaan van Niekerk (2019). "A comparison of object-based image analysis approaches for field boundary delineation using multi-temporal Sentinel-2 imagery". In: *Computers and Electronics in Agriculture* 158, pp. 294–302. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2019.02.009>.
- Whiteside, Timothy G., Guy S. Boggs, and Stefan W. Maier (2011). "Comparing object-based and pixel-based classifications for mapping savannas". In: DOI: doi : 10.1016/j.jag.2011.06.008.
- Willhauck, G., Schneider T., De Kok R., and Ammer U. (2000). "Comparison of object oriented classification techniques and standard image analysis for the use of change detection between SPOT multispectral satellite images and aerial photos". In: *Proceedings of XIX ISPRS Congress, Amsterdam, July 16–22*.
- Wu, Yanyan and Qian Li (2022). "The Algorithm of Watershed Color Image Segmentation Based on Morphological Gradient". In: *Sensors* 22.21. ISSN: 1424-8220. DOI: 10.3390/s22218202.
- Young, Ian (1983). "Image analysis and mathematical morphology, by J. Serra. Academic Press, London, 1982, xviii + 610 p. 90.00". In: *Cytometry* 4, pp. 184–185. DOI: 10.1002/cyto.990040213.

- Zand, Mohsen, Shyamala Doraisamy, Alfian Abdul Halin, and Mas Rina Mustaffa (2016). "Ontology-Based Semantic Image Segmentation Using Mixture Models and Multiple CRFs". In: *IEEE Transactions on Image Processing* 25.7, pp. 3233–3248. DOI: 10.1109/TIP.2016.2552401.
- Zhao, Hengshuang, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia (2017). "Pyramid Scene Parsing Network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhou, W. and A. Troy (2008). "An object-oriented approach for analysing and characterizing urban landscape at the parcel level". In: *International Journal of Remote Sensing* 29.11, pp. 3119–3135. DOI: 10.1080/01431160701469065.
- Zhu, Guobin and Dan Blumberg (2002). "Classification using ASTER data and SVM algorithms;: The case study of Beer Sheva, Israel". In: *Remote Sensing of Environment* 80, pp. 233–240. DOI: 10.1016/S0034-4257(01)00305-4.
- Zhu, Song Chun and A. Yuille (1996). "Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.9, pp. 884–900. DOI: 10.1109/34.537343.

To what extent are field data dispensable for training the ResUNet-a neural network in automatic field boundaries delineation?

Zoé Kempenaers

Accurate field delineation is critical to the implementation of up-to-date and accurate agricultural monitoring systems. Faced with challenges such as limited access to field data and modern technology, developing countries need innovative solutions to ensure effective monitoring of their agricultural systems. The traditional method of manual parcel delimitation, as used in Europe, is proving too costly, time-consuming and error-prone to be applicable in developing countries. In this context, the use of deep learning could meet the needs of these countries, but one drawback is that it requires a very significant volume of field data to be effective. The main objective of this thesis is therefore to train a deep learning model using a classical unsupervised image segmentation algorithm in order to minimise the need for large amounts of field data through the use of transfer learning. This research is carried out in Wallonia, Belgium, taking advantage of the large volume of field data provided by its LPIS. The ResUNet-a neural network is used and trained several times independently, with different types of training and amounts of field data, resulting in a number of different versions of the model. Comparison of the performance of each model shows that pre-training a model with algorithmically generated data can substantially improve performance, but only for small training datasets. The study also demonstrates the resilience of the ResUNet-a model through its ability to perform despite the drastic reduction in the number of labelled fields used for training. The approach studied could therefore be beneficial to all countries that do not have an annual parcel system, enabling significant savings in both manpower and image acquisition costs by reducing the need for manual digitisation based on visual interpretation by up to 90%, with only a slight decrease in model performance. Deep learning proved to be a robust solution, offering automatic feature learning and end-to-end learning capabilities, overcoming the drawbacks of traditional segmentation approaches that rely on manual feature refinement and multiple stages processing. Nonetheless, despite promising results, the approach studied is not necessarily the most effective. In-depth study of deep learning, and in particular transfer learning from a region with abundant field data to another where data is scarce, could help to create more appropriate and effective solutions for providing countries, that do not have an annual parcel system, with accessible decision-making tools in the field of agriculture.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

Faculté des bioingénieurs

Croix du Sud, 2bte L7.05.01, 1348 Louvain-La-Neuve, Belgique | www.uclouvain.be/agro