UCLouvain

AGRO LOUVAIN

Faculté des bioingénieurs

# How does pre-training a Convolutional Neural Network decrease the amount of in-situ data needed to perform agricultural field delineation?

Auteur: Natàlia Dinwoodie Palmés
Promoteur(s) : Dr. Pierre Defourny,
Quentin Deffense
Lecteur(s) : Dr. Sophie Bontemps,
Pr. Xavier Draye
Année académique : 2025-2026
Mémoire de fin d'études présenté en vue de l'obtention du diplôme de Master : SAIV - MSc Erasmus+ GEM : Geo-Information Science and Earth Observation for Environmental Modelling and Management

# Acknowledgements

I want to extend my gratitude to Professor Dr. Pierre Defourny for always adding his expertise with a trained eye. Your support and valuable insights were greatly appreciated throughout the course of this study. I warmly thank my thesis readers and evaluators Dr. Sophie Bontemps and Pr. Xavier Draye for taking the time to review and evaluate this work.

I am eternally grateful to my thesis co-supervisor Quentin Deffense whose guidance and expertise has been essential in completing this research. A big thanks for answering all my questions and passing on your extensive body of knowledge.

A heartfelt thanks to my LLN friends who added brightness and joy to my days during this time. Thank you to my partner for being there for me and always having a way to cheer me up.

A big thank you to my family for supporting me and for always being keen on understanding exactly what it is that I am researching.

# Contributions

The contributors to each stage of this study are found in the table below, with a description of the tasks performed at each stage underneath.

| **Stage** | **Main contributor:** | **Other contributor(s): Quentin Deffense, Pierre Defourny, Boris Nörgaard** | **Artificial Intelligence: ChatGPT** |
|---|---|---|---|
| Conceptualisation | Natàlia D. | Proposition of the research topic by Quentin Deffense and Pierre Defourny. | |
| Methodology | Natàlia D. | Methodological overview and ideation of the *big picture* processing steps by Quentin Deffense.<br>The CNN2D model architecture designed and coded by Quentin Deffense.<br>For the Case Study, Boris Nörgaard provided the in-situ segmentation of field boundaries for Pakistan. | ChatGPT used for writing code and brainstorming some processing steps. |
| Investigation | Natàlia D. | | |
| Formal analysis | Natàlia D. | | |
| Validation | Natàlia D. | Quentin Deffense and Pierre Defourny were involved in visually validating intermediate and final results. | |
| Writing – original draft | Natàlia D. | | |
| Writing – review & editing | Natàlia D. | Feedback and review of text and figures by Quentin Deffense and Pierre Defourny. | ChatGPT was used to revise the original draft to improve the writing. |
| Visualisation | Natàlia D. | | |
| Supervision | Natàlia D. | | |

The research activities carried out in this work included the conceptualisation of the main research idea and formulation of objectives, as well as the design of the methodological framework and selection of appropriate analysis approaches. The work further involved the execution of the research process and data collection, followed by the application of formal analytical techniques to process and interpret the data. The results were subsequently validated through verification and assessment procedures. In addition, an initial manuscript was prepared, followed by iterative revision and editing to refine the both content and structure. The work also included the development of visual materials to effectively present the results. Throughout the project, supervision and coordination of the research activities were maintained, ensuring coherent planning and execution.

The main contributor in the mentioned stages is Natàlia Dinwoodie Palmés. Additional contributions were made by Quentin Deffense and Pierre Defourny who were involved in the conceptualisation, methodology, validation and review and editing stages. Boris Nörgaard collaborated in processing and providing a dataset. Finally, the use of AI tools were employed for the methodological steps of writing source code and for improving original writing. The use of AI tools was exercised responsibly and in compliance with the UCLouvain AGRO Faculty guidelines on GenAI.

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

Remote Sensing (RS)
Machine Learning (ML)
Convolutional Neural Networks (CNN)
Earth Observation (EO)
Ensemble Learning (EL)
Unsupervised Learning (UL)
Random Forest (RF)
Artificial Intelligence (AI)
Artificial Neural Network (ANN)
Rectified Linear Unit (ReLu)
Mean Squared Error (MSE)
Matthews Correlation Coefficient (MCC)
True Positive (TP)
True Negative (TN)
False Positive (FP)
False Negative (FN)
Intersection over Union (IoU)
Area of Interest (AOI)
Training Area (TA)
Evaluation Area (EA)
Near-Infrared (NIR)
Normalized Difference Vegetation Index (NDVI)
Vegetation Index (VI)
Land Parcel Identification System (LPIS)
European Union (EU)
Walloon Public Service (SPW)
Système Intégré de Gestion et de Contrôle (SIGeC)
WALlonie Land Use and Occupation (WALONS)
European Infrastructure for Spatial Information in Europe (INSPIRE)

# Chapter 1

# Introduction

Agriculture plays a central role in global food security, land management, and environmental sustainability. Accurate monitoring of agriculture is essential for supporting crop production, yield estimation, subsidy allocation, and environmental impact assessment. In particular, accurate agricultural information enables the ability to make data-driven decisions to have a more efficient and sustainable agricultural system.

A fundamental requirement for many agricultural analyses is the availability of precise field boundary information. Field boundaries define the regions within which agricultural practices, productivity, and environmental indicators are assessed. However, in many regions, up-to-date and spatially consistent field boundary datasets are either incomplete, outdated, or unavailable. Additionally, obtaining reliable in-situ data on field boundaries is expensive, labour intensive and time consuming. This limitation restricts the ability to perform high-resolution agricultural monitoring and creates a need for automated approaches to perform field delineation particularly in regions which lack this information.

Remote Sensing (RS) combined with Machine Learning (ML) offers a powerful opportunity for deriving agricultural field boundaries from satellite imagery. Particularly, image segmentation methods based on Convolutional Neural Networks (CNNs) have shown strong potential for extracting object boundaries from Earth Observation (EO) data. However, the performance of such deep learning models typically depends on the availability of large amounts of reliable in-situ reference data, which are expensive, time-consuming to collect, and often limited in spatial coverage.

To address this challenge, this study investigates whether pre-training strategies can reduce the need for large in-situ datasets while maintaining accurate field delineation performance. Specifically, the study explores the use of Ensemble Learning (EL) based unsupervised segmentation to generate pre-training data for a CNN, with the aim of improving model performance under limited labelled data conditions. The research focuses on the delineation of agricultural fields in Wallonia, Belgium. This region is small but diverse as both winter and spring crops are grown and overlap in growing seasons, making it an interesting case for accurate field boundary delineation. For this area, detailed and trustworthy field boundary and non-crop area datasets are available and are used here to simulate varying degrees of data scarcity.

Several studies have been carried out on agricultural field boundary delineation using RS data and ML models including CNNs and other segmentation models. Additionally, there are a limited number of studies that have explored the use of Ensemble Learning methods for improving segmentation performance but research in this area remains relatively sparse. Furthermore, only few studies have specifically examined the effect of pre-training on field boundary delineation performance. While some existing studies utilize models with pretrained weights, the impact of pre-training itself is typically not the primary focus of the analysis and is therefore not systematically evaluated.

The UCLouvain's Geomatics Research Lab has researched this topic before. In fact, in 2024, a master thesis by Zoé Kempenaer was performed on field boundary delineation in Wallonia using the CNN model ResUNet-a. The objective was to determine to what extent in-situ field boundary data was dispensable for model training and model performance was evaluated on various datasets of decreasing amounts of in-situ data. Additionally the study compared the performance of a Scratch and a Pretrained model using an unsupervised segmentation, and results indicated that pre-training shows potential in improving model performance especially when in-situ data is limited.

As a result, this research aims to address some of these gaps in literature by investigating the effect that pre-training can have on model performance on datasets of decreasing in-situ data. Specifically, the research will help to answer the question of how pre-training influences the ability of a CNN to maintain accurate field boundary delineation when limited labelled data are available. In doing so, the study seeks to assess whether pre-training represents a potential strategy for improving segmentation performance in data scarce conditions.

# Chapter 2

# Literature Review

## 2.1 Image segmentation

Image segmentation is the procedure of aggregating pixels into particular segments which are intended to represent meaning. These segments can be things like elements of the image, objects, boundaries and more. In this process, a label is assigned to each pixel, where the same label indicates common characteristics of the pixels (Prabu & Gnanasekar, 2021).

RS leverages satellite imagery and other geospatial data to derive information about the Earth's surface and processes. Consequently, image segmentation plays a crucial and growing role in RS applications, facilitating the extraction, classification, and interpretation of spatial features with greater precision and efficiency. Image segmentation is performed using computational algorithms that analyze the spectral, spatial, temporal or textural properties of pixels to group them into homogeneous regions representing distinct land-cover or object classes.

There are two main approaches to image segmentation: semantic segmentation and instance segmentation, which differ in how they classify and distinguish objects in an image.

### 2.1.1 Semantic segmentation

Semantic segmentation is a type of pixel-wise classification in which every pixel in an image is assigned a label corresponding to a class category. In this approach, all pixels belonging to the same class receive the same label, regardless of how many separate objects of that class are present. Thus, semantic segmentation does not distinguish between individual object instances meaning that multiple objects of the same type are treated as a single unified class (Molina et al., 2025). For example, applied to the field of forestry, a semantic segmentation could, classify objects of the same class such as trees, grass, sky, buildings, shrubs, bushes, rocks, steams, and more, which can allow for the monitoring of plant changes, understanding wildlife habits, estimation of forest health and design of forestry methods (Wołk & Tatara, 2024).

### 2.1.2 Instance segmentation

Instance segmentation is a type of image segmentation in which pixels are grouped into individual objects and assigned a unique label per object instance. Unlike semantic segmentation, instance segmentation differentiates between individual objects of the same class, meaning that all pixels belonging to a single object share the same label, while pixels of a different object are assigned a different label, even if they are of the same class. This enables the identification and distinction of each individual object in the image (Molina et al., 2025). In regards to the forestry example, instance segmentation is more suitable for tasks like tree counting, individual tree monitoring, and inventorying the forest (Wołk & Tatara, 2024).

## 2.2 Common segmentation algorithms

A wide range of segmentation algorithms exist, including edge-detection approaches and watershed algorithms.

### 2.2.1 Edge Detection

In an image, sharp spatial variations in pixel values can indicate the transition between distinct classes within an image. Edge detection algorithms aim to identify these abrupt changes which can represent boundaries between objects or classes. Most classical edge detection methods use a kernel, or a weighted matrix of coefficients, which is applied to the image through a convolution operation to extract or enhance specific image features such as edges. The kernel is applied as a moving window across the image, and at each location a weighted sum of neighboring pixel values is computed. The size and weight of the kernel determine what is highlighted in the image, such as horizontal lines, vertical lines, diagonal structures, curves, vertices and more. The different types of operators that can be used include original, Canny, Laplacian, SobelIX, SobelIY, Sobel, Kirsch, Roberts and more (Yu et al., 2023).

Additionally, the various algorithms typically have parameters that can be adjusted which affect the performance of the segmentations. For example, the parameter sigma controls the standard deviation of the Gaussian smoothing filter applied before edge detection which directly controls the amount of noise reduction. A higher sigma value induces more smoothing and less sensitivity to noise, which can lead to smoother edge boundaries but loss of fine details. Another common parameter is the threshold which establishes how strong a change from one pixel to the next must be before a pixel is classified as an edge. Increasing the threshold results in fewer edges detected, as only pixels with a large intensity or gradient changes are classified as an edge, while lower values allow for weaker edges to be detected but may be more susceptible to noise.

#### 2.2.1.1 Sobel filter

The Sobel filter computes the first order derivative of the pixel values in an image which corresponds to the rate of change in pixel intensity. It can detect edges in vertical and horizontal orientations based on the type of kernel applied. The sobel filter kernels include those in Figure 1.

(a)

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

(b)

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure 1: Sobel Filter kernel applied (a) horizontally to detect vertical edges (b) and vertically to detect horizontal edges.

#### 2.2.1.2 Laplacian filter

A Laplacian filter works by calculating the second derivative of the raster image which represents how quickly pixel values change. The assumption is that areas of large change are likely edges or boundaries (Xu & Xiao, 2018).

The filter kernel is passed over the image as a moving window. A typical 3x3 laplacian kernel is shown in the first matrix in Figure 2 (a). Should the kernel pass over a relatively homogeneous area as shown in matrix (b) of Figure 2, the negative center weight (-4) will counteract slight changes in the +1 areas. The operation for the first 3x3 window in bold in matrix (b) is shown in Equation 1. In the kernel (a) the central -4 weight counteracts the four 1 weights resulting in a homogeneous output raster (c).

(a)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

(b)

$$\begin{bmatrix} \mathbf{10} & \mathbf{10} & \mathbf{10} & 10 & 10 \\ \mathbf{10} & \mathbf{10} & \mathbf{10} & 10 & 10 \\ \mathbf{10} & \mathbf{10} & \mathbf{10} & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

(c)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2: Basic Laplacian kernel (a) applied on pixel values of a homogeneous area (b) to produce the resulting raster (c).

An example calculation of basic Laplacian kernel operation on homogeneous area is as follows:

$$(10 * 1) + (10 * 1) + (10 * 1) + (10 * 1) + (10 * (-4)) = 0 \quad \text{(Eq.1)}$$

In an example of a heterogeneous area as is shown in Figure 3 (b), when the same kernel is applied, the negative center weight emphasises the differences from its neighbours. In the resulting raster (c) the edges of the heterogeneous area clearly stand out. An example calculation of this example is shown below.

(a)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 10 & \mathbf{10} & \mathbf{50} & \mathbf{10} & 10 \\ 10 & \mathbf{10} & \mathbf{50} & \mathbf{10} & 10 \\ 10 & \mathbf{10} & \mathbf{50} & \mathbf{10} & 10 \\ 10 & 10 & 50 & 10 & 10 \\ 10 & 10 & 50 & 10 & 10 \end{bmatrix}$$

(c)

$$\begin{bmatrix} 0 & 0 & 60 & 0 & 0 \\ 0 & 0 & \mathbf{60} & 0 & 0 \\ 0 & 0 & 60 & 0 & 0 \\ 0 & 0 & 60 & 0 & 0 \\ 0 & 0 & 60 & 0 & 0 \end{bmatrix}$$

Figure 3: Basic Laplacian kernel (a) applied on pixel values of a heterogeneous area (b) to produce the resulting raster.

An example calculation of basic Laplacian kernel operation on heterogeneous area is as follows:

$$(10 * 1) + (50 * 1) + (50 * 1) + (10 * 1) + (50 * (-4)) = 60 \text{ (Eq. 2)}$$

#### 2.2.1.1 Canny filter

The Canny is known to be highly efficient in detecting edges within images and is one of the most used algorithms for this task. As a first step, as described by Agrawal and Desai (2024), the image is smoothed by a Gaussian filter in order to remove noise and suppress non-significant detail. Following, Sobel filters are used to compute image gradients in the vertical and horizontal directions from which the gradient magnitude and gradient direction are obtained. Furthermore, non-maximum suppression is then performed where pixel values perpendicular to gradient direction are compared to their neighbors, and if their pixel value is less than edge pixels, their value is suppressed to zero but if the pixel value is greater, it is recomputed as the new edge. This results in thin and well defined edges. Finally, hysteresis thresholding is applied, where an upper and lower threshold are used to identify legitimate edges. To do this, pixels with strong gradient magnitudes above the upper threshold are directly classified as strong edges. Pixels with magnitudes between the thresholds are considered weak edges and are only retained if connected to a strong edge, in which case they are immediately classified as a strong edge. This process helps to keep connectivity along edges and form continuous edge contours (Agrawal & Desai, 2024).

### 2.2.2 Watershed

The watershed algorithm treats an input image as a topographic surface where pixel values are interpreted as elevation. The algorithm finds local minima and establishes their surrounding or influence area as a catchment basin. Watershed lines are the boundaries between the different basins. The algorithm simulates flooding of the basins starting from the minima and gradually filling the basins. When water from two basins is about to merge, a watershed line is drawn to prevent merging which defines the boundary between regions. The flooding continues until all pixels are assigned to a basin or watershed and results in a segmentation based on local minima and topography (Wen et al., 2022).

The watershed algorithm includes the parameter sigma that controls the amount of smoothing applied to the image before segmentation. A higher sigma value results in a smoother image surface by reducing local variations and noise.

## 2.3 Ensemble Learning

EL is a technique that can be used with deep learning models to improve model performance. The *building blocks* of EL are baseline classifiers, also known as weak learners, base learners, base classifiers or component models. From here on they will be referred to as Unsupervised Learning (UL) algorithms. They are defined as the individual models that are applied on input data and output a prediction. Individual outputs are then combined using an aggregation function to generate a single prediction. The aggregated collection of component models is known as the ensemble. UL algorithms are typically simpler or weaker models that, on their own, may have limited predictive power, but when combined through ensemble techniques, they collectively achieve higher accuracy and better generalization (Yang, Lv, & Chen, 2023; Mohammed & Kora, 2023). EL techniques and methodologies have been studied extensively. For example, Anwar et al. (2014) created a global optimization ensemble model to predict on 7 datasets of different parameters and achieved increased classification accuracies from 1% to 30% compared to running singular models.

Two techniques exist for training individual baseline classifiers, it can be done sequentially or in parallel.

### 2.3.1 Sequential ensemble methods

In the sequential ensemble method, the first baseline classifier is trained on input data, and the following models are trained on the performance of the previous model in the chain. Each consecutive model tries to improve the classification, based on the errors that were made previously by the models that came before (Jadama et al., 2024). This aims to correct errors after each run, and ultimately obtain one final classification result with a minimal error (Mohammed & Kora, 2023).

### 2.3.2 Parallel ensemble methods

In the parallel method, the various baseline classifiers are trained simultaneously and independently, on the input data and a result is extracted from each. Then, all of the classification outputs are combined for a final prediction (Jadama et al., 2024). The principle of this technique relies on the independence between classifiers in that the errors between models are different and get averaged out when all the final results are aggregated (Mohammed & Kora, 2023).

## 2.4 Ensemble Learning aggregation methods

Various EL aggregation methods have been developed and refined over time. The most popular ones include bagging, boosting and stacking.

### 2.4.1 Bagging

Bagging, also called bootstrap aggregation, is a parallel aggregation method that involves dividing the training data into multiple bootstrap datasets and training an individual model on each dataset. The predictions from these independently trained models are combined to produce one single, more robust prediction. This technique is advantageous over an individual model as it leverages the diversity among models trained on different subsets of the data, thereby reducing the likelihood that the ensemble overfits to specific samples or noise present in the full training set. By training on varied subsets rather than the entire dataset at once, bagging increases model robustness and improves generalization performance (Zhang et al., 2022b). Bagging has been shown to reduce bias and variance and can help mitigate overfitting. Using bagging in early iterations of baseline classifier runs typically reduces bias, and in later iterations, it reduces variance. A disadvantage is that bagging is not optimal for data with high noise levels (Anwar et al., 2014).

### 2.4.2 Boosting

Boosting as an aggregation method is described as a method to transform poorly performing UL algorithms into better performing algorithms. To begin, all input data is assigned an equal weight, and a UL algorithm is made to output a prediction. The weights of the incorrectly classified samples are increased, and the following weak learner is made to try to correct the previous mistakes (Zhang et al., 2022b). In the initial iterations this is a bias-reducing method,

and in later runs variance is reduced (Anwar et al., 2014). This process is repeated sequentially, meaning one learner at a time, until all the learners have produced an output. Finally, the results of all runs are aggregated through a weighted vote for classification and a weighted sum for a regression (Zhang et al., 2022b).

### 2.4.3 Stacking

Stacking differs from bagging and boosting in the sense that the predictions of several poor performing UL algorithms (level 0 models) are not aggregated directly, instead they are used as inputs into a meta-learner (level-1 model), which learns how to best combine them to improve the final prediction (Zhang et al., 2022b). Literature suggests that the performance of a stacking algorithm is dependent on the accuracy and diversity of the algorithms, with diversity being defined as the level of complementariness between algorithms. Ideally, harnessing the predictive ability of UL algorithms with high accuracy and diversity leverages the diversity between highly performing UL algorithms to maximize generalization accuracy (Zhang et al., 2022a). While adding more UL algorithms may increase computational cost and memory requirements, better model performance is not guaranteed. Empirical studies suggest that stacking three to five UL algorithms often offers a good balance between accuracy and efficiency (Zhang et al., 2022a).

## 2.5 Ensemble Learning for boundary detection

Literature on the application of EL specifically for field boundary extraction is scarce. In particular, studies investigating EL as a strategy for either direct boundary delineation or as a pretraining approach for boundary detection are largely absent.

Mei et al. (2022) demonstrated that the Mask R-CNN algorithm, a convolutional neural network, applied to satellite imagery can successfully delineate smallholder field polygons, achieving F1 scores greater than 0.72. However, the models produced different types of mis-delineations depending on the input image type. To address these inconsistencies, the authors suggest using an ensemble approach that combines predictions from multiple models to improve overall delineation accuracy.

## 2.6 Random Forest Algorithm

The Random Forest (RF) model is a well known and one of the most commonly used algorithms in ML. With ML being described as a branch of Artificial Intelligence (AI) in which algorithms learn patterns from data to perform tasks such as prediction and classification without being explicitly programmed. Random Forest utilises multiple decision trees to generate predictions. Each decision tree is trained using a random subset of the training data and a random selection of input features. The trees begin from an initial decision node and progressively split the data into branches based on feature threshold until a final classification or prediction is reached. Then, the algorithm ensembles the outcomes of multiple decision trees through a voting mechanism to produce the final prediction. The advantage of using multiple decision trees to ensemble the final prediction, is the achieved better generalization ability than a single decision tree. Additionally, using multiple decision trees introduces variability, reduces overfitting and increases model robustness (Sun et al., 2024).

In Figure 4, a representation of the RF algorithm is displayed. As can be seen the full dataset is divided into various training subsets from each of which the data is used to produce a decision tree. The outcomes are finally ensembled.

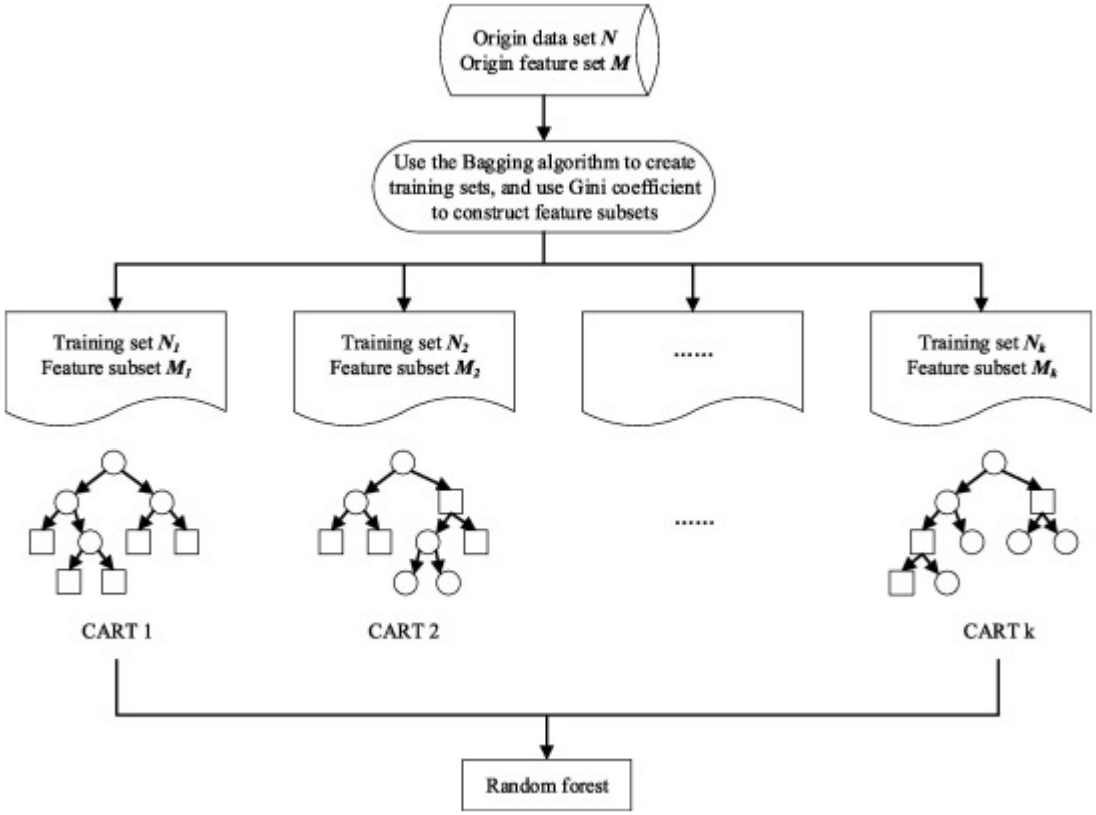



Figure 4: Representation of a Random Forest Algorithm (Sun et al,. 2024).

## 2.7 Artificial Neural Network (ANN)

The concept of an ANN was first introduced in 1942 by Warren McCulloch and Walter Pitts with their artificial neural network model inspired by the structure and functions of biological neurons (Pires, Santos, & Pereira, 2023). The principle was rooted in the ability for a biological neuron to activate or not activate, which was translated into an ANN structure capable of processing binary logic. A big breakthrough was made in 1958 with Frank Rosenblatt's new type of ANN called a Perceptron, which could detect input-output relationships without explicitly being programmed to do so (Pires, Santos, & Pereira, 2023). From then on, this type of model has continued to be developed with many advances occurring from the 1960s to modern day.

The ANN learns the relationships between variables in a dataset allowing them to solve complex problems with high accuracy, including those lacking explicit algorithmic solutions. Deep learning models such as ANNs typically require large datasets, a high quality labels dataset, substantial memory resources and a significant computational time for training.

### 2.7.1 Architecture of an ANN

The Architecture of an ANN can be broken down into an input layer, hidden layers and an output layer.

#### 2.7.1.1 Input layer

The input layer consists of the raw input features of the dataset (Mienye & Swart, 2024). In this layer, no computations are performed, the input values are simply passed forward to the next layer of the, which is typically the first hidden layer (Montesinos-López, et al., 2022).

#### 2.7.1.2 Hidden layer

Hidden layers are where the computations and transformations from information received from previous layers take place (Mienye & Swart, 2024). The neurons within these layers can be organized and interconnected in various formulations, influencing the overall architecture or topology of the network. The number and types of these layers are considered hyperparameters, which must be defined prior to training as they affect the model's capacity and performance. Within these layers, the network performs a series of learned transformations which are stored in the weights and biases and that enable it to extract increasingly abstract and relevant features from the data (Montesinos-López, et al., 2022).

#### 2.7.1.3 Output layer

This is the final layer of the neural network by which its neurons output the model's final predictions. Neurons in the output layer receive the processed features from the preceding hidden layers and translate them into an output. The output is a representation of the model's understanding of the input data (Li, 2024; Montesinos-López et al., 2022).

### 2.7.2 Components of an ANN

Artificial neural networks consist of interconnected computational elements, such as neurons, layers, weights, biases, activation functions, and loss functions that collectively define their structure and learning behavior.

#### 2.7.2.1 Neuron

Neurons are a set of connected units that make up the layers of the ANN. They receive a set of inputs, which are combined with a set of weights and an activation function to produce one single output. Neurons are connected to each other within and between layers. The connections or links between neurons carry a weight that is learned which, when combined with the input data, will either activate or suppress the neuron. An active neuron will pass information forward, while a suppressed neuron will not. This output is fed into the next set of neurons in the following layer.

#### 2.7.2.2 Layers

The ANN is composed of various types of layers. A layer is a collection of neurons at a certain depth of the ANN that execute a computation on the input data received by the neurons.

#### 2.7.2.3 Weights

The connections between neurons are called weights. These weights represent the strength of influence one neuron has on another. Initially, random weights are assigned to all the linkages when the algorithm is initialized. During training, these weights are adjusted to optimize the network's performance, meaning that this parameter is trainable (Islam et al., 2019). After training, the optimized weights represent the learned structure of the data, effectively storing the information acquired during the learning process.

#### 2.7.2.4 Bias

The bias term is often added after the product of the inputs and weights. It represents the intercept or threshold of a neuron (Montesinos-López, et al., 2022). Its function is to offset the result, which shifts the input of the activation function and gives the model more flexibility to activate under different conditions. This is a parameter that is learned and optimized by the model during the process of training and learning (Montesinos-López et al., 2022).

#### 2.7.2.5 Activation function

This is the mathematical operation applied after the neuron input is multiplied by the weight which will determine the output of the neuron. This hyperparameter's purpose is to classify the input to see if the neuron is activated and passes on its information (Li, 2024). Commonly used activation functions include linear, Rectified Linear Unit (ReLu), sigmoid, softmax and tanh functions (Montesinos-López, et al., 2022). The use of nonlinear activation functions has enhanced the ability for deep learning models to model complex nonlinear relationships, thus providing more stable and efficient models (Li, 2024).

### 2.7.3 Structure of an ANN

Figure 5 illustrates a simplified schematic of an (ANN) with two hidden layers, where each white circle is a neuron containing associated weights, a bias, an activation function, and an output. Each input is multiplied by its corresponding weight, and the weighted inputs are summed together with the bias term. This combined value is then passed through an activation function, which introduces non-linearity and produces the neuron's output. There are many other more complex ways neurons can be connected; this is known as the topology of a neural network. In this case, Figure 5 represents a unilayer with a univariate output, meaning one layer of trainable neurons predicting one variable as output (Montesinos-López, et al., 2022).

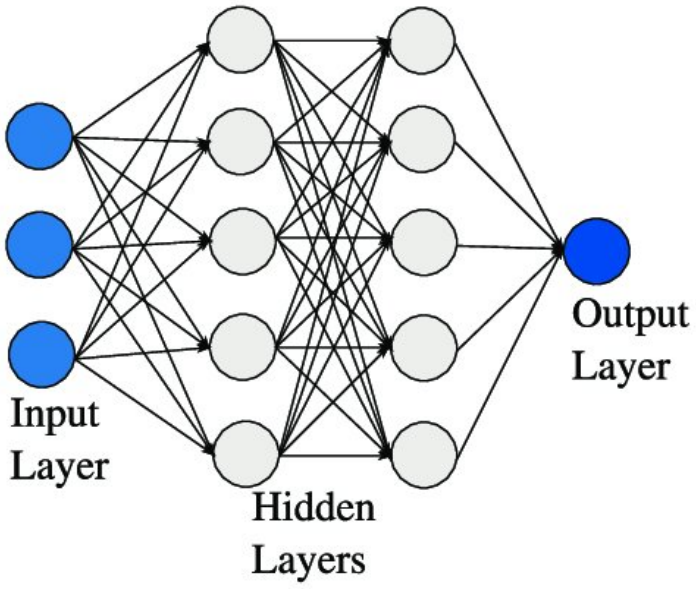



Figure 5: Structure of an ANN (Ataei et al., 2020).

The generic equation to represent the mathematical model for one neuron is shown in Equation 3. *Xi* represents the input into the neuron, Wi is the weight outputted by the neuron, *b* is the bias and *f* is the activation function. The output of the neuron is y (Li, 2024). A mathematical model for neurons can be written as follows (Li, 2024):

$$y = f\left(\sum_{n}^{i=1} w_i \cdot x_i + b\right) \quad \text{(Eq. 3)}$$

### 2.7.4 Training an ANN

When training an ANN, the first step is to split the data into the calibration and testing datasets.The *calibration dataset* is the portion of the data used to develop the model, including fitting the network parameters and tuning hyperparameters, while the *testing dataset* is held out entirely and used only at the end to provide an unbiased assessment of the model's performance.

The calibration dataset is further split into the *training* and *validation* subsets, where the training subset is used to update the model's weights, and the validation subset is used to monitor performance during training and help prevent overfitting. A representation of this can be seen in Figure 6.

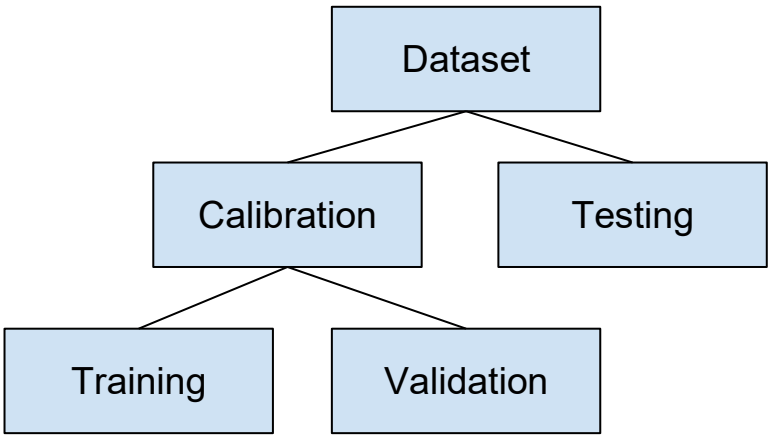



Figure 6: Overview of Dataset Partitioning in Machine Learning.

In the process of training an ANN, the transmission of information from the input to output layer is known as feedforward propagation (Li, 2024). The information flows in a forward manner, following the architecture of the ANN through the links between neurons and layers. To begin the training process, the weights of the network are assigned random numbers from 0 to 1, after which the model is run (Sekhar & Sai Meghana, 2020). Once the output layer produces the predictions, these predicted values are compared to the actual values, and the difference between them is quantified using the loss function. The loss function measures how distant the ANN's predictions deviate from the true values, serving as an indicator of the model's predictive error (Montesinos-López, et al., 2022).

During training, minimizing this loss becomes the primary optimization objective, whereby the model iteratively adjusts the weights and biases as these are the variables that influence the loss (Montesinos-López, et al., 2022). This is known as backpropagation, where the weights and biases of the neural network are recalculated based on the gradient of the loss function (Li, 2024). By summarizing the prediction errors across the entire dataset into a single scalar value, the loss function guides the optimization algorithm toward parameter values that improve predictive accuracy (Montesinos-López, et al., 2022). In regression problems, Mean Squared Error (MSE) is a suitable metric for calculating the loss function, and in a classification problem, cross-entropy is apt (Li, 2024). However, the choice of loss function is not fixed and depends on the specific task and problem formulation.

In machine learning, an epoch refers to one complete pass through the entire training dataset during the training process, where every sample has been used once to update the model's parameters (Hussein & Shareef, 2024). Each epoch is typically divided into smaller subsets of the data called batches, where a batch is a fixed number of training samples processed together before the model's parameters are updated. After each batch, the loss function is backpropagated throughout the neural network from the output layer to the input layer and the model's weights and biases are updated. Then the model is trained on the next batch with the updated weights and biases.

It is commonly experienced and expected, that as the number of epochs increase, the training loss decreases due to the model progressively refining its parameters to better fit the training data, and eventually stagnates when the model has extracted nearly all learnable patterns

from the data  Along with this, the validation loss initially tends to decrease as the model learns generalizable patterns, but it will begin to increase once the model starts memorizing the training data rather than learning from it. At the point where training loss is low, and before validation loss starts to increase is where training should stop in order to prevent overfitting (Hussein & Shareef, 2024; Miseta et al., 2024). This is represented in Figure 7 showing when to ideally stop training an ANN to prevent under and overfitting.

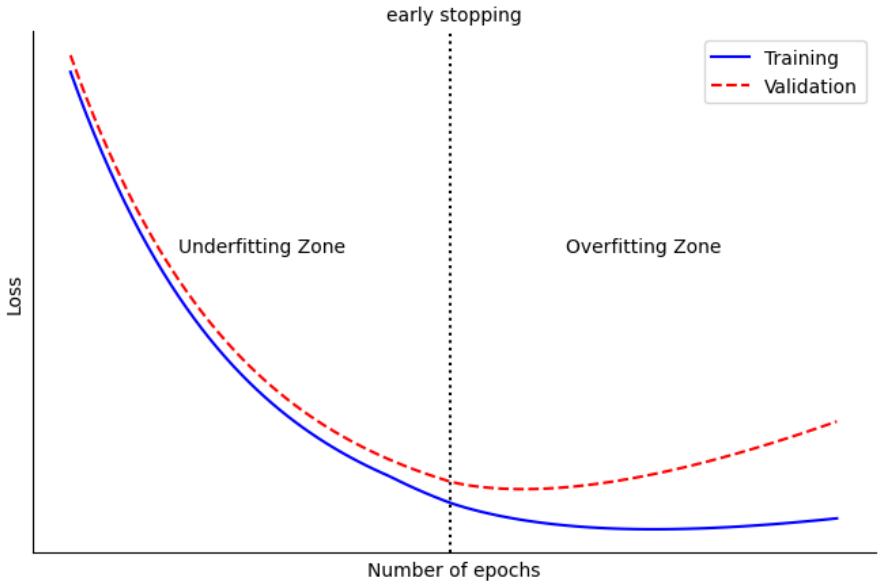



Figure 7: Representation of Training and Validation loss evolution over epochs during CNN training.

Learning for an ANN is understood as the iterative process of adjusting the network's weights and biases to minimize the loss function (Montesinos-López, et al., 2022). This optimization enables the network to perform the task more accurately by finding the set of parameters that produce the smallest possible error (Islam et al., 2019).

## 2.8 Pre-training an ANN

The concept of transfer learning was first introduced in a research paper by S. Bozinovski and A. Fulgosi in 1976 but became more well known in the 1990s (Bozinovski, 2020). Transfer learning is based on the foundation that knowledge acquired while solving one problem can be reused to improve learning performance on a related task. In deep learning applications, this is typically achieved by transferring learned network parameters from a trained source model to a target model operating within a similar application (Han et al., 2021). The learned parameters vary depending on the transfer learning technique, and can include the initialization weights, shared feature patterns, the partial network structure and generalized feature representations (Tan et al., 2018). By utilizing knowledge acquired during prior training, transfer learning reduces the need to train a target model entirely from random initialization. This can speed up training, and circumvent two of the major bottlenecks in deep learning which are the massive amounts of training data required and insufficient training data available (Tan et al., 2018).

Transfer learning is the precursor of modern day pre-training. Pre-training was revolutionized by Hinton, G. E., Osindero, S., and Teh, Y.-W. in their 2006 research paper publication which demonstrated a novel way of pre-training a very deep neural network one hidden layer at a time prior to training (LeCun, Bengio, & Hinton, 2015). A pre-training procedure was utilized which involves creating layers of features without the use of labelled data. These pseudo-inputs mimic or reconstruct the raw data inputs. Subsequently, the model is pre-trained on the pseudo-inputs. The learned weights of the pre-training are used to initialize the weight parameters for the model training on the raw data inputs (LeCun, Bengio, & Hinton, 2015).

The effects and efficiency of pre-training varies between applications, models and implementation. He, Girshick, and Dollár (2019) find that pre-training a CNN can accelerate model training during the early stages, reducing the number of epochs required to reach similar performance. However pre-training does not necessarily improve final accuracy and in some cases shows no benefit compared to training from Scratch when sufficient training data is available. However, when little training data is available, pre-training may still provide a useful initialization that helps stabilize early learning and improve training efficiency (He, Girshick, & Dollár, 2019). Furthermore, Pu et al. (2023) suggest that pre-training primarily benefits models that fit the data poorly by improving the optimization process which means achieving faster reduction in training and validation loss. On the other hand, models that already fit the data well may gain little or no advantage from pre-training. Additionally, when the model has sufficient fitting ability, pre-training mainly improves convergence speed and enables faster generalization during training. Convergence speed refers to the number of epochs required for the loss to stabilize and faster generalization means earlier and more stable improvement in validation loss relative to training loss (Pu et al., 2023).

As mentioned previously, the ANN is configured to have a certain network structure with weights that are initialized randomly, as this is information that the model does not know yet. During training, these parameters are iteratively learned in order to minimise loss and allow the model to recognize meaningful patterns. In the case that for a certain application, it is the pre-trained weights that are transferred from the pre-training to the training, this means that model training will not start from random initialization, but by a more informed starting point.

## 2.9 Convolutional Neural Network

CNNs are a type of ANN distinguished by their use of convolutional layers. These layers perform convolution operations to effectively extract spatial features from input data**.** They have demonstrated high performance in diverse tasks such as image classification, image segmentation, image retrieval, object detection, image captioning, face recognition, pose estimation, traffic sign recognition, speech processing and more (Bezdan & Džakula, 2019). Designed to process single or multi-channel array data, also referred to as tensors or input channels, these models operate on multidimensional numerical grids or arrays. In the context of remote sensing, a single or stack of remotely sensed images can be regarded as one such array.

The architecture of a CNN involves an input layer and a series of convolutional and pooling layers explained below. A representation of a CNN structure can be seen in Figure 8.

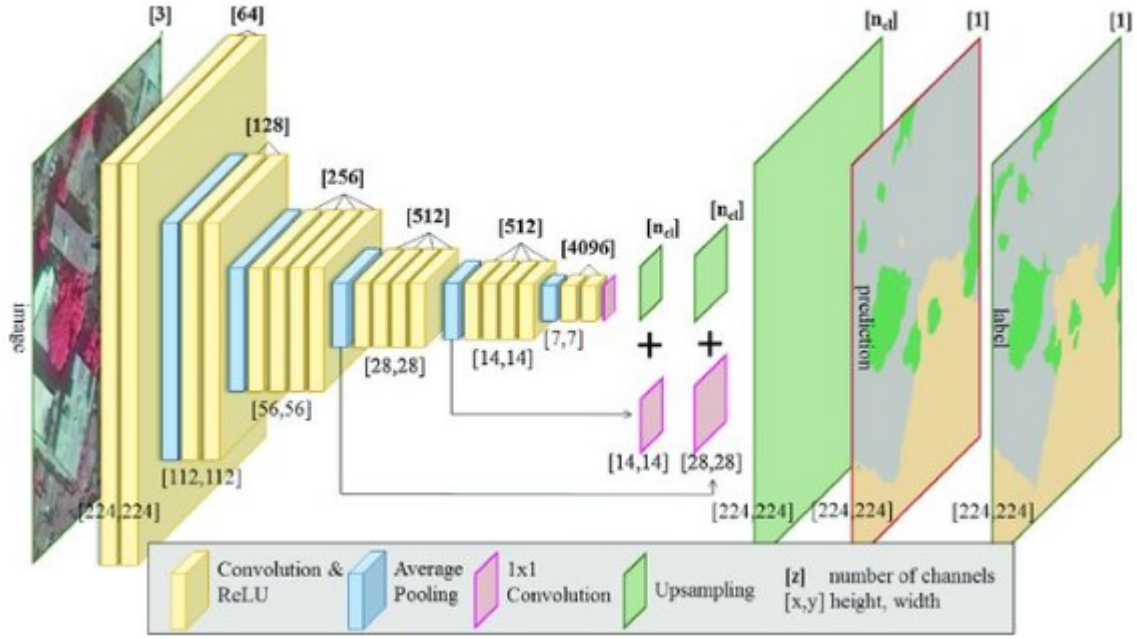



Figure 8: Example of structure of a CNN for semantic segmentation (Wurm et al., 2019).

### 2.9.1 The Convolutional Layer

In the convolutional layer, a series of filters are passed over the entire extent of the input array, performing an operation at each position. A filter is applied as a kernel, which acts as a moving window of fixed size. Common kernel sizes include 3×3, 5×5, and 7×7 pixels. The moving window is typically placed at the top-left corner of the input array and made to slide over the image with a defined step. Each kernel contains a weight matrix that is learned during training, allowing it to highlight or detect specific features such as edges, textures, or patterns within the input data. The dot product of the value of the input image and the kernel weight at the overlapping position is mapped onto an output layer, at the same position. A non-linear activation function can be applied to the operation, to capture non-linear trends in the data. After the filter operation is performed at one position, the window shifts by a predefined step called the stride, applying the same operation at the new location. The stride determines how far the window moves across the input array after each operation. The resulting outputs are called feature maps and contain the output of the filter operation at each pixel position (Bezdan & Džakula, 2019).

### 2.9.2 The Pooling Layer

It is common to have pooling layers inside a CNN architecture after convolution layers. These layers are responsible for downsampling or reducing the spatial dimensions of the feature maps produced by the convolutional layers. A filter, often called a pooling window, is passed over the input feature map in a similar sliding manner as in convolution, but instead, it applies a summary operation to the values within the window. Typical values for filter size are 2×2

pixels, and stride in pooling layers are 2. This means that the pooling window covers a 2×2 pixel area and moves two steps at a time, effectively reducing the feature map's width and height by approximately half in this particular case (Bezdan & Džakula, 2019).

Max pooling is a common type of pooling summary operation, where the maximum value within each window is selected as the output, highlighting the strongest feature response in that region. Another option is average pooling, where the average of all values within the window is computed, providing a smoothing effect by summarizing the overall presence of features. In this way, pooling layers reduce the size of the output feature map, lowering the spatial resolution but preserving the most prominent or average information from the original features. This reduction helps the network to focus on the most important aspects of the data (Bezdan & Džakula, 2019).

## 2.10 CNN2D

A CNN2D is a type of CNN which is designed to process two-dimensional data. The model is constructed in such a way to preserve the spatial relationships between pixels, and is particularly appropriate for remote sensing applications (Ding et al., 2023). This type of model is apt for performing classification and segmentation tasks.

A typical CNN2D model takes inputs in terms of image height, image width and number of channels. Then, the model can contain various layers and components to apply to those inputs and subsequent processing, whose configuration will depend on the exact application of the model. These include the input layer, convolutional layers activation functions, pooling layers, flattening layers, fully connected layers and the output layer (Nouna et al., 2025). The CNN architecture is composed of convolutional blocks with residual-style connections and batch normalization. This allows for hierarchical feature extraction while preserving gradient flow. During model training, model performance can be tracked over multiple epochs, and early stopping can be applied to prevent overfitting. A CNN2D can produce predictions for multiple classes or output multiple target variables, depending on the task.

## 2.11 Image segmentation evaluation metrics

In object segmentation, there are two main types of evaluation metrics that provide insightful information on model performance. These evaluation metrics are run on the *testing area* predictions (see Figure 6) or final prediction area in order to assess the agreement between the predicted segmentation outputs and the corresponding ground-truth labels. Pixel based evaluation metrics, measure how accurately each pixel is classified at an individual pixel level. On the other hand, object-based metrics evaluate how well predicted objects match reference objects in terms of overlap, shape and overall spatial agreement.

### 2.11.1 Pixel-based metrics

An example of a commonly used pixel-based metric is the Matthews Correlation Coefficient.

#### 2.11.1.1 Matthews Correlation Coefficient (MCC)

The MCC is a pixel-based evaluation metric which compares the actual values and predicted values of a classification. Positive samples correspond to instances belonging to the class of interest, whereas negative samples correspond to instances that do not belong to that class. In the context of field boundary prediction, actual values are the field boundaries, while the predicted values are field boundary model predictions (Chicco & Jurman, 2020). Predicted values can be classified as true positives, true negatives, false positives, or false negatives depending on whether the predictions correctly or incorrectly match the ground-truth labels. Figure 9 represents a confusion matrix of these terms.

*True Positive (TP)*: Actual positive values correctly predicted as positive values. For a field boundary prediction this is a field pixel correctly predicted as a field.

*True Negative (TN)*: Actual negatives are correctly predicted as negatives. This would be a non-field pixel correctly predicted as non-field

*False Positive (FP)*: Actual negatives are incorrectly predicted as positive values. Therefore, this occurs when a non-field pixel is incorrectly classified as a field.

*False Negative (FN)*: Actual positive pixel values are incorrectly predicted as negatives. This means a non-field pixel incorrectly predicted as a field.

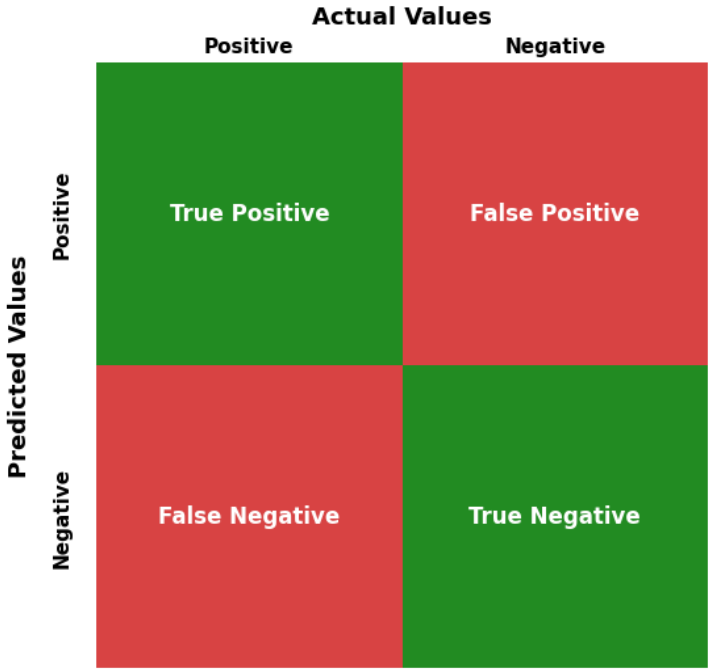



Figure 9: Predicted and Actual Values Confusion Matrix.

This metric is a reliable and widely used evaluation metric, particularly in ML applications. In contrast to other evaluation metrics, this measure is not strongly affected by class imbalance, thereby avoiding oversimplistic evaluation performance assessments driven primarily by the classification of the majority class (Chicco & Jurman, 2020). MCC values range from +1 to -1, where +1 signifies perfect classification and -1 means perfect misclassification. The denominator normalizes the metric using the size of both positive and negative classes, which is why MCC remains balanced even when the classes are unevenly distributed. The MCC equation is defined as:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \quad \text{(Eq. 4)}$$

### 2.11.2 Object based metrics

The Intersection over Union is a widely used object-based metric.

#### 2.11.2.1 Intersection over Union (IoU)

The IoU is an object based metric which ascertains the similarity between two datasets. In the context of object segmentation, this metric determines the similarity in area between the actual and predicted objects. The formula used to calculate the IoU is defined as (Padilla et al., 2020):

$$J(B_p, B_{gt}) = \text{IOU} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} \qquad \text{(Eq. 5)}$$

Each predicted and actual object is compared based on their respective geometry. The predicted and actual geometries are paired based on a criteria such as the objects with the greatest overlap. Then, the overlapping area between the corresponding predicted object $B_p$ and the actual or ground truth object $B_{gt}$ are divided by the union between them. This is illustrated in Figure 10. Furthermore, a threshold is set for which if the IoU of a set of objects is greater than the threshold, it is considered as a correct detection, and if the IoU is below the threshold, it is an incorrect detection. The IoU has an established range of values from 0 to 1, where 0 signifies no overlap between the predicted and actual objects, and 1 is perfect overlap (Padilla et al., 2020).

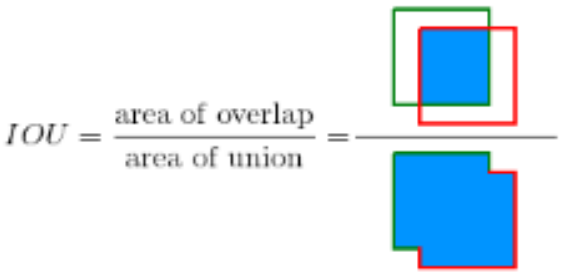



Figure 10: IoU Diagram (Padilla et al., 2020).

# Chapter 3

# Objective

## 3.1 Research Question

Field boundary detection is important for accurate land monitoring, crop management, productivity assessment and resource allocation, and can accurately be performed using image segmentation. To carry out this application using deep learning, it is necessary to use large amounts of in-situ data which are expensive and can be difficult to obtain.

This study aims to answer the following question:

*To what extent does pre-training a CNN using EL affect model performance compared to a non-pretrained Scratch model when using decreasing amounts of in-situ data to delineate the boundaries of agricultural fields in Wallonia?*

The research question will be answered by completing the following steps:

1. Apply UL segmentation algorithms (approx. 18 - 22) on monthly modal NDVI composites using various inputs and parameters to obtain varying preliminary predictions of field boundaries.
2. Use an ensemble of UL segmentations to obtain Extent, Boundary and Distance layers of each segmentation run to pre-train a Convolutional Neural Network.
3. Prepare Extent, Boundary and Distance for a Scratch model which does not use pre-training.
4. Run the Pretrained and Scratch model using decreasing amounts of in-situ data (100%, 50%, 10%, 5%, 4%, 3%, 2%, 1%) to obtain a vectorized delineation of field boundaries in Wallonia.
5. Evaluate the performance of each model run using pixel-based metrics based on actual and predicted values, and object based metrics based on the area overlap of actual and predicted objects. Prediction and evaluation results will be used to determine the minimum amount of in-situ data required to maintain high segmentation accuracy.
6. Perform a Case Study to determine the transferability of the established methodology and the effect of pre-training in a region with limited in-situ data such as Pakistan.

# Chapter 4

# Materials and Methods

The methodology employed during this research is explained in this chapter with explanations of each processing step.

## 4.1 Area of Interest (AOI)

The AOI is the region of Wallonia in Belgium. Wallonia is a relatively small region, containing approximately 16,900 km². Despite its size, it is quite diverse in terms of its agricultural landscape. The region is a particularly interesting AOI as winter crops and spring crops are cultivated, which have distinct growing seasons that overlap during the year. This seasonal complexity increases the challenge of accurate field boundary delineation.

Within this AOI, all the stages of the workflow are carried out, including generating datasets for model training and evaluation, as well as model prediction and validation of the generated outputs. Moreover, Wallonia is mainly covered by six Sentinel-2 tiles namely: T31UES, T31UFS, T31UGS, T31UER, T31UFR and T31UGR. These tiles represent the tiles with the greatest spatial overlap with the region and are therefore used to define the study area extent. As a result, marginal areas of Wallonia that fall outside these tiles are not included in the analysis. Figure 11 below indicates the AOI and tiles included within it.

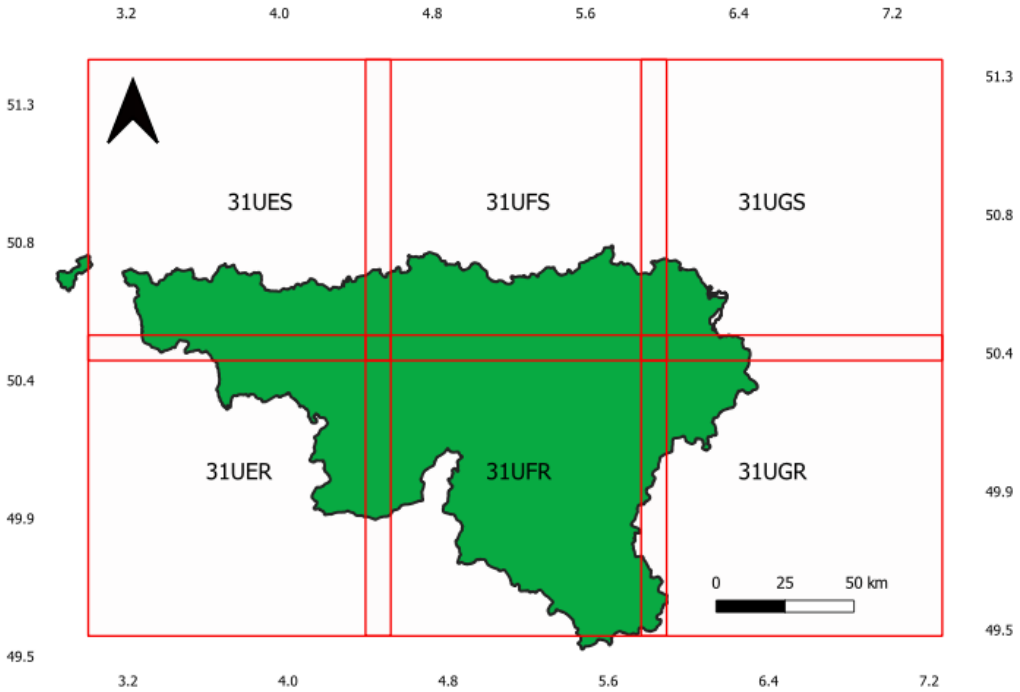



Figure 11: Area of Interest with Sentinel-2 tiles included.

Three areas have been selected as training areas in order to ensure both variation and representativeness in terms of terrain and agricultural practices of the border study area. Training Area 1 (TA1) is located across the Brabant and Hennuyer silty plateau, Training Area

2 (TA2) covers the Hesbaye area and Training Area 3 (TA3) is situated in the Ardennes region. These areas are chosen to capture a range of landscape and agricultural conditions present within Wallonia, thereby improving the generalisability of the model.

Inside each training area, a respective validation area (VA1, VA2, VA3) is defined, representing approximately 10% in both area and in-situ data of its respective training area. Evaluation areas are reserved for model performance assessment during calibration and are excluded from the training.

Furthermore, the calibrated models will be made to predict on the entire Wallonia AOI. Prediction evaluation will also be made on the entire extent of the AOI. The training and evaluation areas are shown in Figure 12.

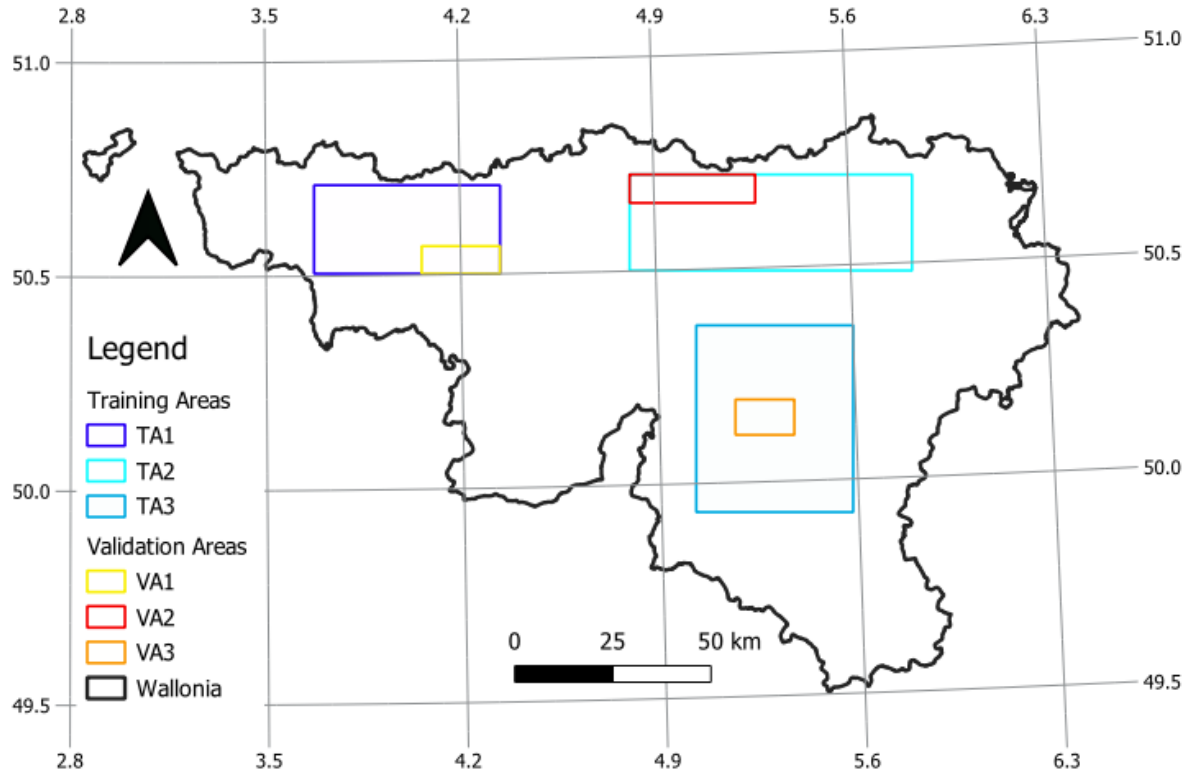



Figure 12: Training and Evaluation Areas in the AOI.

## 4.2 Study Period

The study period considered for this study spans the year 2024, from January 2024 to December 2024. The year 2024 was a particularly cloudy year which poses limitations for a RS analysis. Consequently, not all months can be included as input and prediction data for the CNN due to the presence of clouds. Only the satellite acquisitions for the months of April, May, June, July, August, September and October could be retained. This means that satellite data is not included for the season of winter and a reduced amount of data is included for the season of autumn. This is not ideal as the inclusion of a full annual cycle would be preferable to capture seasonal variability.

However, since the primary objective of this study is to compare the performance of two modeling approaches, being the Pretrained and Scratch methodologies, the inputs into the two models are the same. Therefore, the impact of reduced temporal coverage is expected to affect both approaches equally, preserving the validity of the comparative analysis.

In Wallonia in 2024, 45% of land is agricultural land and about 33% of land is forests. Within the quantity of agricultural land, 41% are meadows and pastures while the remaining 59% are winter are spring crops (Service public de Wallonie, n.d.). Some of the main crops that are sown are winter wheat, maize and potatoes. Spring crops like potato are sown in April and emerge in May, they flower in July and are harvested in September and October. Similarly for silage maize, sowing takes place in April-May, emergence in late May, flowering in August and harvesting in September-October. Winter crops like winter wheat are sown in October, flower in June and are harvested in July-August (Gobin et al., 2023).

At sowing time, the soil is most often bare and is visually observed as brown, which becomes green during emergence, remains green during flowering, turns light green at tuber setting and harvesting occurs when the field is dark green (Gobin et al., 2023). Therefore, the months of April to October capture the major phenological stages of the crops which correspond to reflectance variations throughout the growing season.

In Figure 13, an orthophoto from WalOnMap is shown, developed by the Walloon Public Service (SPW) at 25cm precision for 2024 (Bassine et al., 2020). The exact date that the image was taken is not known, but the WalOnMap 2024 orthophoto acquisitions were taken on April 6, 2024 and between August 10 and September 21, 2024. Therefore, the date of the orthophoto can be in mid-spring or early autumn. On top of the orthophoto the LPIS fields considered in the study (explained in section 4.3.2) are shown for the different crops in the area. As can be seen, at the time of the image there are varying degrees of crop emergence ranging from bare soil (no emergence) to full emergence as seen by varying and deepening colors of green. Generally all cereal and industrial crops parcels are bare soil while fresh vegetables and root crops range from bare soil to emergence, probably depending on crop type.

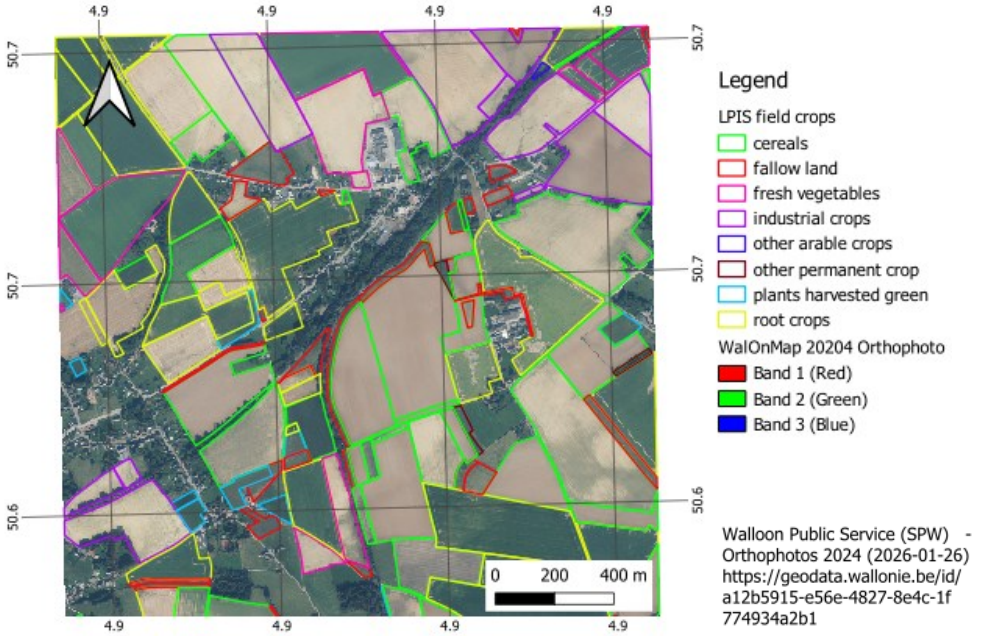



Figure 13: LPIS field crops on WalOnMap 2024 Orthophoto.

## 4.3 Data Sources

### 4.3.1 Satellite Acquisitions: Sentinel-2

The Sentinel-2 sensor is chosen for this study due to its combination of high spatial resolution and frequent temporal revisit. The 10-m spatial resolution of the selected spectral bands enables the detection of detailed spatial changes. In addition, the revisit time of approximately five days allows for the monitoring of temporal dynamics in vegetation. This combination of spatial and temporal characteristics makes Sentinel-2 well suited for capturing both structural and phenological variability in agricultural environments. The Sentinel-2 mission operator and data provider are respectively the European Space Agency and the EU Copernicus program. Images were acquired from the Geomatic research lab of UCLouvain.

The Sentinel products obtained are from the MAJA481 collection and have a processing level of L2A. This means that acquisitions have been atmospherically corrected to provide Bottom-Of-Atmosphere reflectance. The bands considered are B04 and B08. Additionally, the MAJA481 collection has corresponding cloud masks for its acquisitions, however these were not used. Instead, the Fmask46 considering 0pct is used, due to its reputation for better performance. This Fmask is an automated algorithm that detects cloud, cloud shadows and snow, and classifies them (Claverie et al., 2018). Pixel values that are 0 in the mask represent valid pixels, therefore when the Fmask is applied to Sentinel-2 acquisitions, the remaining pixels that are valid observations.

Spectral information on the selected bands is displayed on Table 1.

Table 1: Spectral Information of Sentinel-2 bands used in the study (European Space Agency, 2015)

| Band | Band Name | Wavelength (μm) | Resolution (m) |
|---|---|---|---|
| B04 | Red Band | 0.64 - 0.68 | 10 |
| B08 | Near-Infrared (NIR) | 0.85 - 0.88 | 10 |

#### 4.3.1.1 Normalized Difference Vegetation Index (NDVI) composites from Sentinel-2

The NDVI is one of the most commonly used Vegetation Index (VI) for vegetation monitoring and detection. This VI utilizes reflectance values from the NIR and red spectral bands. Vegetation typically exhibits strong absorption in the red region due to chlorophyll content and high reflectance in the NIR due to internal leaf structure. This distinct spectral contrast allows NDVI to effectively distinguish vegetated surfaces from other land cover types with different spectral responses in these wavelength ranges.

The index is normalized and can produce values ranging from -1 to 1. Wherein values below 0 represent non vegetated surfaces such as clouds, snow, rocks, water and artificial materials.

NDVI values from 0 to 0.2 represent open soil, values from 0.2 to 0.5 signify sparse vegetation and high NDVI values from 0.7 to 1.0 indicate dense vegetation (Korchagina et al., 2020).

The equation for NDVI is depicted in Equation 6 below. Regarding Sentinel-2 bands, the $\rho_{NIR}$ is represented by band B08 and $\rho_{red}$ is represented by band B04. The equation to calculate NDVI is expressed as (Asam et al., 2023):

$$NDVI = \frac{\rho_{NIR} - \rho_{red}}{\rho_{NIR} + \rho_{red}} \quad \text{(Eq. 6)}$$

In RS analyses, it is often necessary to process data over extended temporal periods. This can mean processing many acquisitions and requiring vast amounts of resources. In such cases, NDVI composites are commonly generated by aggregating NDVI layers over certain temporal intervals within the study time period, in order to balance computational efficiency with the need to capture meaningful temporal dynamics. These composites integrate NDVI observations from multiple acquisition dates into a single representative value per pixel over a given time window. The length of this temporal window is typically selected based on the objectives of the study and the dynamics of the target system, with monthly aggregation being a commonly used approach.

The choice of compositing methods has a strong impact on the final NDVI product (Asam et al. 2023). For example, median compositing is commonly used when working with high spatial resolution data like Landsat or Sentinel-2 to generate gap-free mosaics over long time periods.

A study by Asam et al. (2023) comparing 13 NDVI compositing approaches including maximum value compositing, median compositing, and multi-criteria weighted approaches, with the aim of generating the most consistent long-term NDVI time series found that there is no one universal optimal approach. However, it was found that median based compositing achieved the best equilibrium between noise reduction, avoiding saturation and maintaining consistency across the spatial and temporal domains (Asam et al. 2023).

### 4.3.2 Land Parcel Identification System (LPIS)

The LPIS is a geographic database overseen by the European Union (EU) as part of the Common Agricultural Policy (CAP) to ensure comprehensive and comparable data across the EU. It consists of an agricultural parcel map of each country. The information provided is an anonymized vector file containing the boundaries of the field and the main crop grown on it. In the region of Wallonia, collecting this information is overseen by the SPW. Farmers have to declare their parcel information such as their crop boundaries for that year, and the crops that will be sown by the 31st of May in the Système Intégré de Gestion et de Contrôle (SIGeC). The published data by the SPW is made publicly available (European Commission, n.d.). The year of data that is used is 2024. The agricultural field boundaries serve as the crop boundary in-situ data for this project.

### 4.3.3 WALlonie Land Use and Occupation (WALOUS)

The  WALOUS is a mapping project which produced a land use map for the region of Wallonia for the year 2018 with 25 cm precision and an updated land use legend and classification (Bassine et al., 2020). It was necessary for this dataset to be produced in order to have a land use classification that was compliant with the European Infrastructure for Spatial Information in Europe (INSPIRE) directive. The project was replicated for the year 2023 and the new land cover map was published in the beginning of 2025 (Service public de Wallonie, n.d.) .

For this study, the WALOUS dataset is acquired for the year 2023, as it is not available for 2024. This dataset is used to extract the non-crop land areas within the AOI. The assumption is that the non-crop polygons will largely remain the same from one year to the next and even many years in the future.

## 4.4 Models and experimental setup

The processing workflows for the Pretrained model and Scratch model methodologies are explained to give a general overview of processing steps. The individual processing steps are explained in more extensive detail in the *4.5 Methodology* section.

### 4.4.1 Scratch Model

For the Scratch model, only the training methodology is applicable. Figure 14 shows the workflow for the Scratch models. As a first step, the six NDVI median monthly composites are created and serve as model Features. Then, in-situ data for crop and non-crop areas from the LPIS and WALOUS sources are used to create the Extent, Boundary and Distance Labels. The Extent characterizes the crop, non-crop and unknown areas while the Boundary contains the crop boundaries and non-crop areas. Finally the Distance is the normalized distance of how far away a pixel is from a crop boundary. The in-situ data are used to create datasets of decreasing percentages of in-situ data shown in blue. For each Training dataset percentage Labels and Features are extracted respectively. Following this, each training dataset is trained by a CNN model and saved. Each saved model is made to predict on the extent of Wallonia using the same six NDVI composites as prediction features. Finally, the LPIS field areas are used to evaluate the accuracy of predictions.

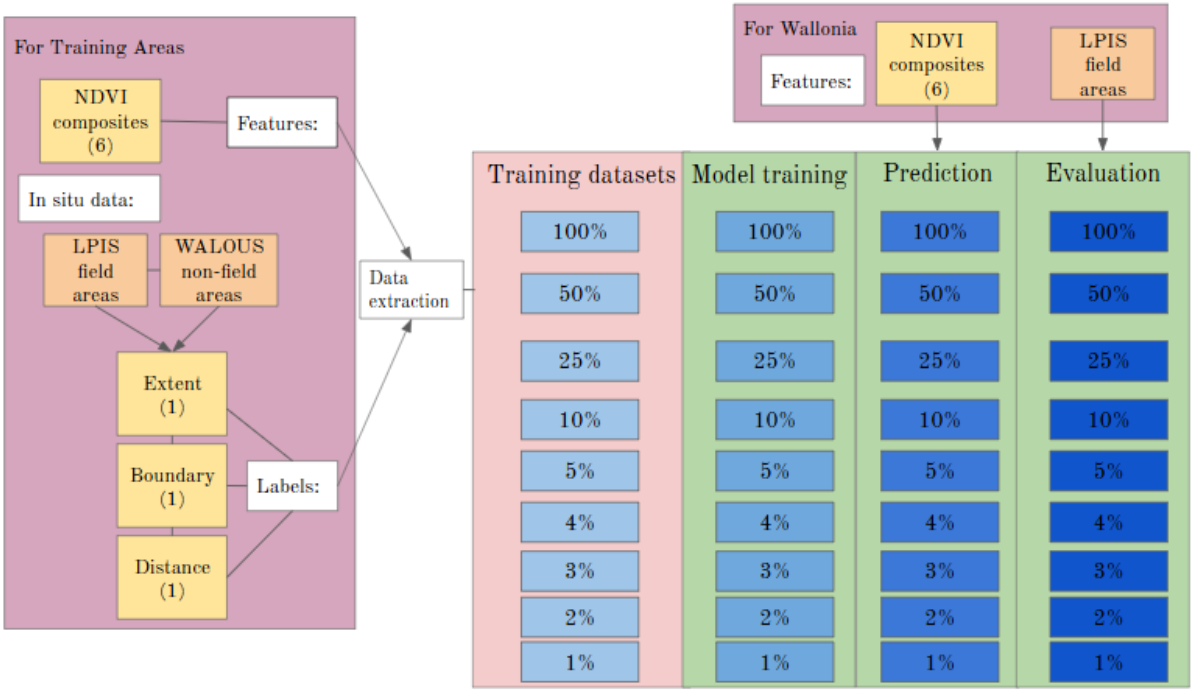



Figure 14: Detailed Workflow of Scratch Model.

### 4.4.2 Pretrained Model

The Pretrained model includes the pre-training and training methodology. The workflow diagram of the Pretrained models is illustrated in Figure 15.

Firstly, in Figure 15 in the pre-training block, pseudo-labels are generated that mimic the inputs that the model will see during the training. The NDVI composites for the six months used are utilized to create a Crop/Non-Crop mask which serves as the Extent pseudo-label. Then, UL algorithms are run on the NDVI composites to produce 22 UL Segmentations which delineate boundaries all over Wallonia. The previously created Extent mask is used to clip each UL Segmentation to Crop areas in order to obtain 22 Boundary pseudo-label layers for boundaries in crop regions. Furthermore, the Distance pseudo-label is obtained for each Boundary layer by calculating the distance each pixel is from a crop Boundary. In the pre-training dataset, the three pseudo-labels make up the Labels while the six NDVI composites compose the Features.

Pre-training is performed in patches due to memory and processing constraints where the extent of Wallonia is subdivided into smaller image patches. The pre-training dataset is then used to train the model, with patches randomly shuffled prior to input to reduce the effect and bias of input order into the model. Pre-training is performed and the Pretrained model is saved to be used in Training. Subsequently, Training is performed in the same way as described in sub-section 4.4.1 for each in-situ data percentage, with the exception that the models are initialized with the pre-training weights.

Pre-training

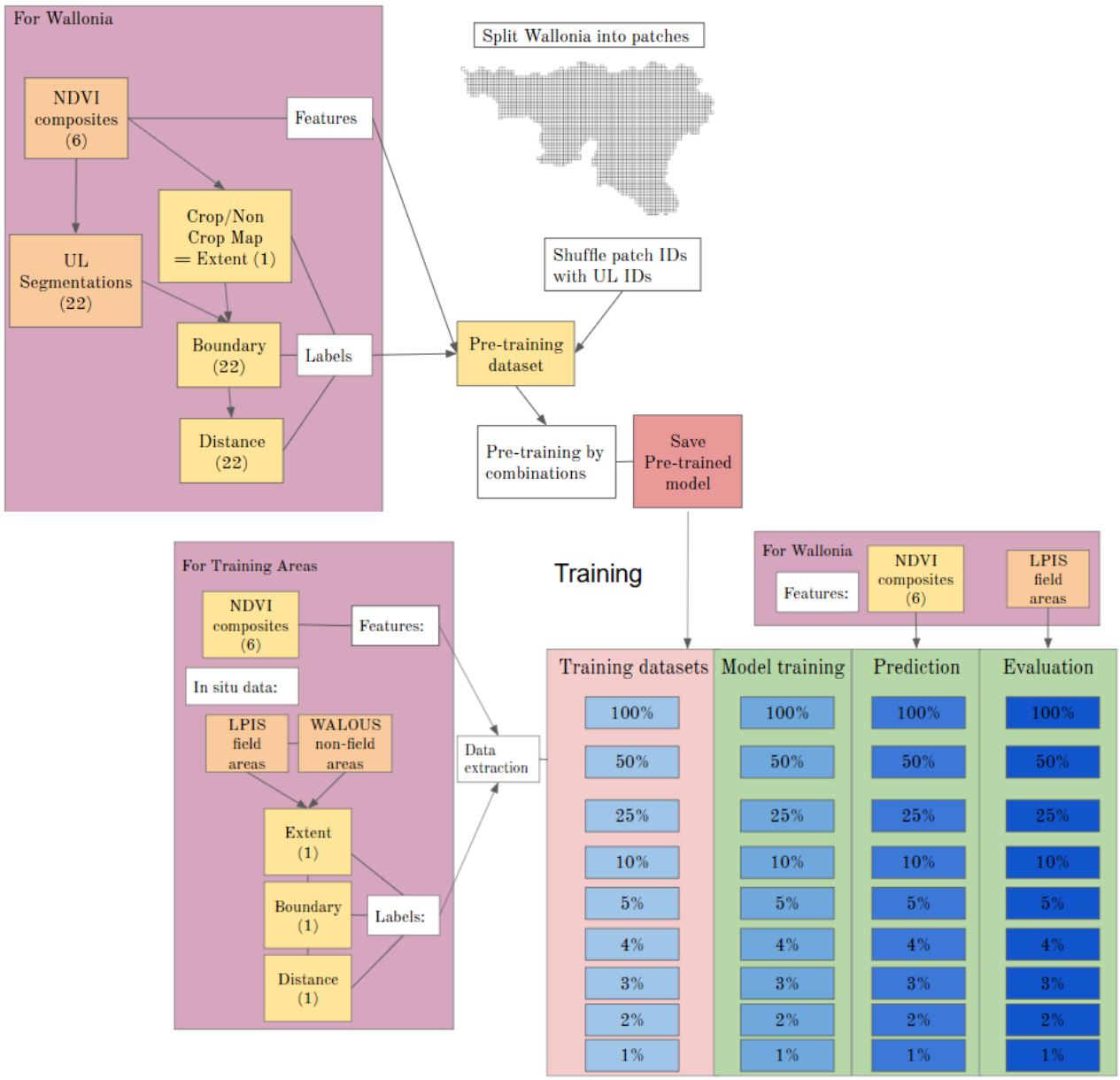



Figure 15: Detailed Workflow of Pretrained Model.

## 4.5 Methodology

Each processing step which composes the methodology of both models is described in detail in this section.

### 4.5.1 Median NDVI composites

Across the 6 tiles considered in the study area, for each month in 2024, images with more than 15% valid pixels within Wallonia in the Fmask are considered. The Fmask is applied to bands B04 and B08, keeping only valid pixels.

For the processing step of running UL segmentation algorithms, the monthly median NDVI composites for each of the 12 months are utilised. However, for the model training only the composites for the months of June, July, August, September and October, monthly median NDVI composites are used. A bi-monthly composite of March-April is generated to diminish the amount of cloud gaps in both months. The months of January, February, March, November and December of 2024 are discarded due to lack of valid data due to clouds.

The calculated monthly median NDVI composites in a selected region of Wallonia are shown in Figure 16 for months July, August and October. The changes in vegetated area and vegetation intensity can be appreciated across the different plots of land throughout the growing season.

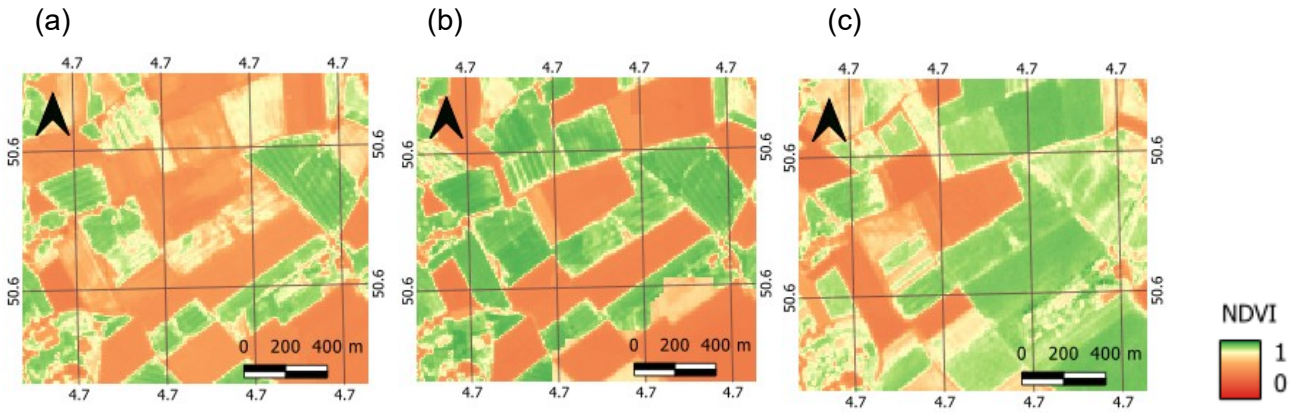



Figure 16: Sentinel-2 Monthly Median NDVI Composites of Months (a) July, (b) August and (c) October for 2024.

### 4.5.2 LPIS pre-treatment

The LPIS dataset is cleaned so as to only include agriculture fields by excluding from the dataset pastures and meadows, permanent grassland, rough grazing and nurseries. The resulting dataset includes 161,740 agriculture fields comprising 4,391.5 km² of the declared agriculture fields in 2024 in Wallonia.

### 4.5.3 WALOUS pre-treatment

The WALOUS dataset for 2023 is obtained, and the non-crop polygons are selected to obtain the working dataset.

### 4.5.4 Reduction of in-situ datasets

Using the cleaned LPIS dataset containing only agriculture fields, the in-situ datasets are created. Only the agriculture fields within the extent of the three training areas are considered. The total number of crop fields are counted, and the field polygons are randomly shuffled. First, the 100% in-situ data subset is created considering all crop field polygons within the training areas and serving as a reference. The datasets of progressively decreasing proportions of in-situ data include 100%, 50%, 25%, 10%, 5%, 4%, 3%, 2%, 1%. For each subsequent fraction of in-situ data, a specified number of polygons, which is calculated as a percentage of the total dataset size, is randomly selected without replacement from the indices of the previous subset. This approach ensures that each smaller subset is fully contained

within the larger one. As a result, nine in-situ datasets of decreasing quantity of crop polygons are obtained of the following percentages.

The same procedure is performed on the non-crop polygons from the WALOUS dataset. Thus, nine subsequent datasets of decreasing size of non-crop polygon in-situ data of the following percentages are obtained: 100%, 50%, 25%, 10%, 5%, 4%, 3%, 2%, 1%.

In Table 2, the number of fields which are included in each in-situ dataset are shown.

Table 2: Corresponding Number of Fields in each in-situ Dataset

| In situ dataset (%) | Number of fields |
|---|---|
| 100 | 47,021 |
| 50 | 23,510 |
| 25 | 11,755 |
| 10 | 4,702 |
| 5 | 2,351 |
| 4 | 1,880 |
| 3 | 1,410 |
| 2 | 940 |
| 1 | 470 |

### 4.5.5Unsupervised Learning (UL) Segmentations

As a previous step to obtaining the pre-training inputs, a rough estimate of boundaries within the area of Wallonia are obtained. These are necessary as a way to give the model an estimation of where boundaries are and what they look like without using in-situ data.

They are obtained by passing segmentation algorithms over the previously created NDVI composites that are designed to find and segment boundaries. The algorithms segment boundaries in a general way, without being able to distinguish between crop, non-crop and other areas. In total, 22 UL segmentation runs are made for the entire extent of Wallonia using different algorithms, parameters and combinations of NDVI images serving as inputs. Each is saved individually for further processing.

The objective of obtaining multiple UL segmentations is for the strengths and weaknesses of each UL classifier to be compensated and balanced out by each other, and to obtain a relatively realistic prediction of the field boundaries in Wallonia. In order to obtain classifications showing variation between them, the algorithms are run on different parameters and inputs. The parameters being those adjustable settings particular to each algorithm, and

inputs being the NDVI composite from different months. Furthermore, the algorithms perform a boundary detection per month that is inputted for each particular run. Then, a voting threshold is applied, where a boundary is retained in the final output only if it is detected in more months than the specified threshold. The various combinations of algorithms, parameters and inputs are shown in Table 3.

Table 3: Runs of Unsupervised Learning Segmentations

| Algorithm | Input Months | Parameters |
|---|---|---|
| Canny Filter | ALL MONTHS (12) | Sigma: 2<br>Voting Threshold: 0.4 |
| | ALL MONTHS (12) | Sigma: 3<br>Voting Threshold: 0.4 |
| | MAY-JUNE-JULY-AUGUST (4) | Sigma: 2<br>Voting Threshold: 0.4 |
| | MAY-JUNE-JULY-AUGUST (4) | Sigma: 3<br>Voting Threshold: 0.4 |
| | AUGUST (1) | Sigma: 2<br>Voting Threshold: 0.4 |
| | AUGUST (1) | Sigma: 3<br>Voting Threshold: 0.4 |
| Sobel Filter | MAY-JUNE-JULY-AUGUST (4) | Threshold: 0.05<br>Voting Threshold: 0.4 |
| | MAY-JUNE-JULY-AUGUST (4) | Threshold: 0.08<br>Voting Threshold: 0.4 |
| | MAY-JUNE-JULY-AUGUST (4) | Threshold: 0.1<br>Voting Threshold: 0.4 |
| | ALL MONTHS (12) | Threshold: 0.05<br>Voting Threshold: 0.4 |
| | ALL MONTHS (12) | Threshold: 0.08<br>Voting Threshold: 0.4 |
| | ALL MONTHS (12) | Threshold: 0.1<br>Voting Threshold: 0.4 |
| Laplacian Filter | MAY-JUNE-JULY-AUGUST (4) | Sigma: 1<br>Threshold:0.02<br>Voting Threshold: 0.4 |
| | MAY-JUNE-JULY-AUGUST (4) | Sigma: 2<br>Threshold:0.01<br>Voting Threshold: 0.4 |

| | | |
|---|---|---|
| | MAY-JUNE-JULY-AUGUST (4) | Sigma: 3<br>Threshold:0.005<br>Voting Threshold: 0.4 |
| | MAY-JUNE-JULY-AUGUST (4) | Sigma: 4<br>Threshold:0.005<br>Voting Threshold: 0.4 |
| | ALL MONTHS (12) | Sigma: 1<br>Threshold:0.02<br>Voting Threshold: 0.4 |
| | ALL MONTHS (12) | Sigma: 2<br>Threshold:0.01<br>Voting Threshold: 0.4 |
| | ALL MONTHS (12) | Sigma: 3<br>Threshold:0.02<br>Voting Threshold: 0.4 |
| | ALL MONTHS (12) | Sigma: 4<br>Threshold:0.005<br>Voting Threshold: 0.4 |
| Watershed | MAY-JUNE-JULY-AUGUST (4) | Sigma: 1<br>Voting Threshold: 0.4 |
| | MAY-JUNE-JULY-AUGUST (4) | Sigma: 3<br>Voting Threshold: 0.4 |

To exemplify, in Figure 17, two obtained UL segmentation outputs are shown, namely the outputs for the Canny filter including all months with a sigma of two, and the Watershed filter for the summer months with a sigma of one. The variability of the outputs can be appreciated.

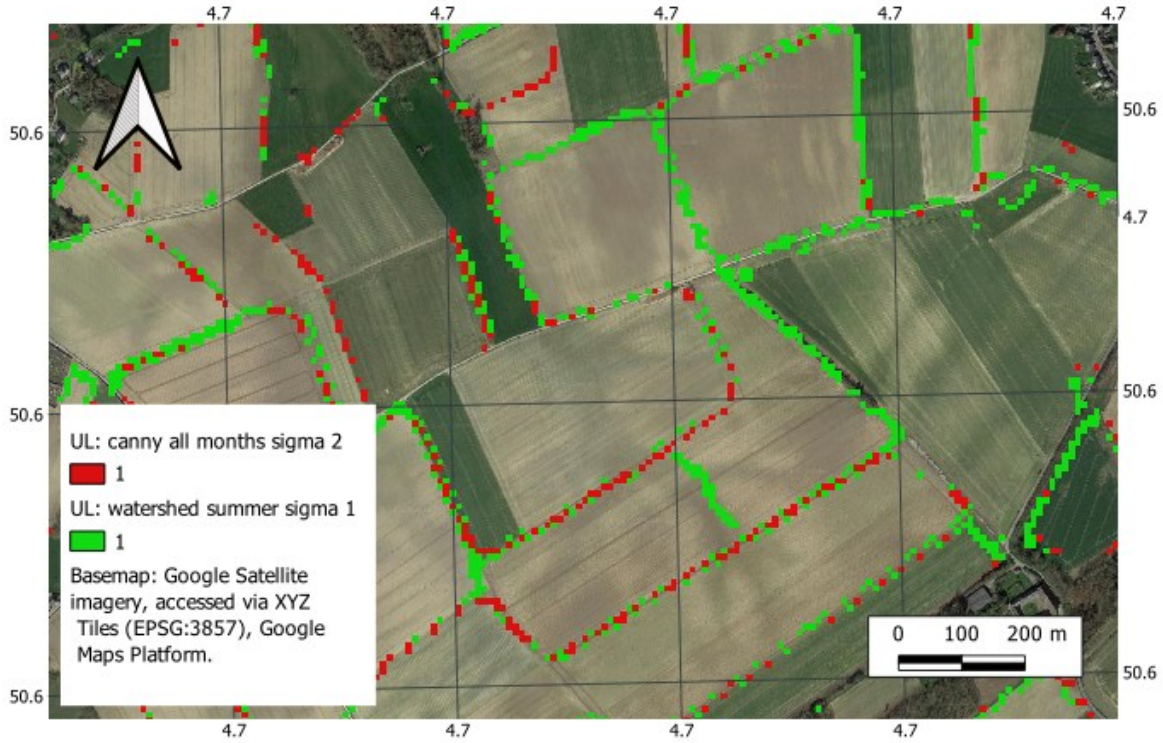



Figure 17: Example of two UL Segmentation Outputs.

## 4.5.6 Pre-training inputs

### 4.5.6.1 Extent Layer - Crop/Non Crop Classification

To generate the pre-training data, it is necessary to identify the locations of agricultural land within the pre-training area. Since field boundary delineation is only relevant within crop areas, a Crop/Non-Crop mask is first created to distinguish agricultural land from non-agricultural land. The Crop/Non-Crop mask is obtained using a ML classification model trained on a subset of labeled spectral data to classify pixels as either crop or non-crop.

The model used is the Random Forest Classifier. To obtain the labels for the model, the 1% in-situ datasets of crop and 1% in-situ datasets of non-crop areas that are previously created are used. The pixels that fall within crop and non-crop polygons are used as the labels, while monthly median composites for months June, July, August, September, October and a bi-monthly composite for months April-May are used for the features. The classification is performed on the entire extent of Wallonia. The model prediction is validated on 100% of the LPIS and WALOUS in-situ datasets for all of Wallonia, resulting in an Overall Accuracy of 0.92. In total, approximately 83.1 million non-crop pixels and 35.0 million crop pixels are correctly classified. Around 7.9 million crop pixels are misclassified as non-crop, while approximately 2.5 million non-crop pixels are incorrectly classified as crop.

The Crop/Non-Crop Classification serves as the Extent Layer to be used in the CNN with the classes of Crop and Non-Crop. The resulting Crop/Non-Crop classification is found in Figure 18.

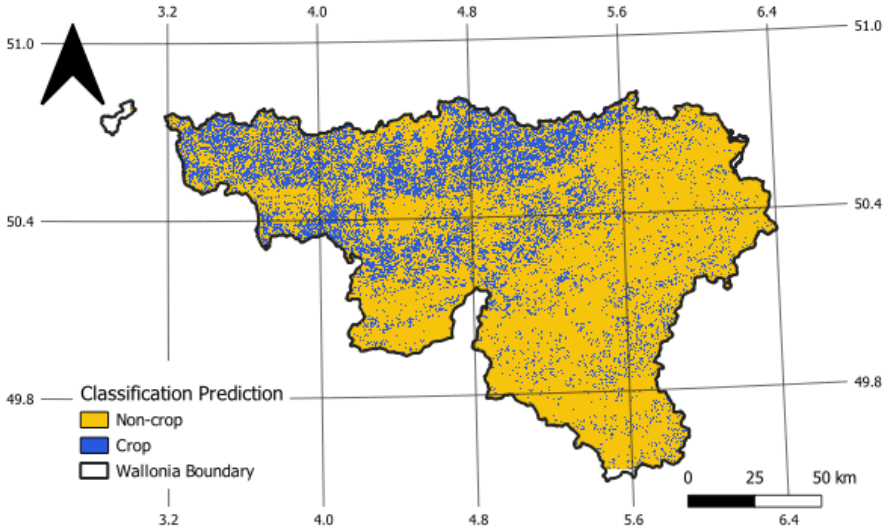



Figure 18: Crop/Non-Crop Classification of Wallonia for 2024 using 1% in-situ data

#### 4.5.6.2 Boundary Layer

To construct the Boundary Layer used for the pre-training, the previously created UL segmentations and Crop/Non-Crop Classification are used. First, the 22 UL segmentation outputs are clipped to the extent of the crop areas in the Crop/Non-Crop Classification ensuring that only boundaries around crop field areas are kept, while boundaries in non-crop areas, like built-up areas, are removed.

However, this strict clipping can lead to fragmented or partially removed field boundaries along crop edges. To mitigate this, a constrained dilation step is applied to the boundary layer prior to masking. This dilation expands boundary pixels by 1 pixel when the boundary is within a crop area. As a result, boundary information is allowed to slightly extend to neighbouring pixels only if boundaries are within valid crop areas. This preserves continuity of field boundaries while preventing leakage into non-crop regions. Finally, from each UL Segmentation, 22 crop Boundary Layers are obtained.

#### 4.5.6.3 Distance Layer

From the 22 UL crop Boundary Layers, the Distance Layers are computed for each. For each Boundary layer, the Euclidean distance is calculated for each pixel position to its nearest crop boundary. The Euclidean distance provides a continuous spatial measure of proximity, ensuring that each pixel is assigned the shortest straight-line distance to a boundary, which makes it robust to irregular shapes and varying boundary geometries. The distance values are then normalized by a fixed maximum distance, converting them into a scale between 0 and 1, where values close to 0 indicate proximity to a boundary and values close to 1 indicate increasing distance from any boundary. The result of this step is 22 respective Distance Layers, one for each of the UL Classifications.

#### 4.5.6.4 Extracts of Pre-training Inputs

A subset of each pre-training input layer is displayed in Figure 19.

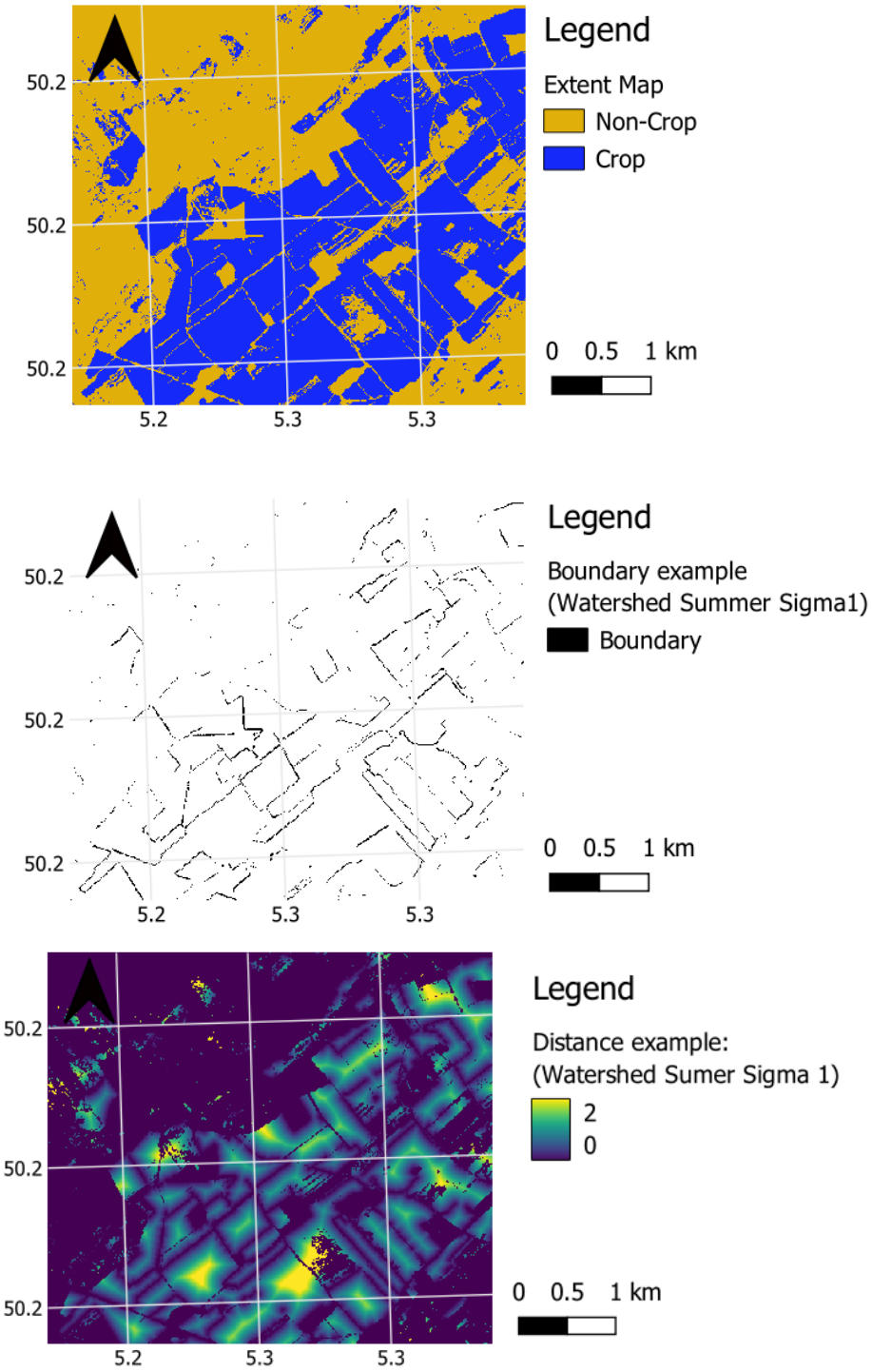



Figure 19: Extent, Boundary and Distance Inputs for Pre-Training.

### 4.5.7 Pre-training

The pre-training is performed with the objective of obtaining the weights and biases with which to initialize the training. The labels for the pre-training are derived from the Extent, Boundary and Distance layers made specifically for pre-training the model and that will differ from the Extent, Boundary and Distance used for training. The same CNN configuration is used as will be used for the training, however, the CNN is only trained on one epoch of data.

As a very large amount of data is used which accordingly requires an excess of resources, the pre-training is executed in patches. Instead of loading all patch information to pre-train the model, the model performs pre-training on a certain number of combinations of patches at a time. First, the region of Wallonia is split into non-overlapping 256 x 256 pixel patches each with a unique ID. Patches with more than 30% cloudy pixels are discarded. This produces 2,182 valid patches over Wallonia. Then, the UL segmentation predictions are assigned a unique ID. A process of randomly shuffling patch IDs with UL segmentation IDs is performed so that 100 combination files are filled with randomly paired patch IDs and UL segmentation IDs, with each patch–segmentation combination assigned only once across all files.

Subsequently, the combination files are processed iteratively. For each combination file, on a patch by patch basis, labels are extracted and corresponding features are derived from the six NDVI monthly and bi-monthly composites. The NDVI composites have previously been normalized using a percentile-based scaling (1st–99th percentile) to produce values between 0 and 1. This results in a corresponding paired label and feature dataset for each configuration file for better model interpretation. The model is then pre-trained by iteratively loading each dataset pair, performing training and proceeding to the next set until all combination datasets have been processed. The model is run for 1 epoch yielding a set of learned initial parameters, which are subsequently stored and used to initialise the Pretrained model.

### 4.5.8 Training inputs

The pre-processing for the training is only computed for the three training areas. It is performed individually for each of the in-situ datasets that decrease in size.

#### 4.5.8.1 Extent Layer

To first generate the Extent Layer, the crop and non-crop in-situ datasets of the particular percentage that is being processed are utilized. Pixels corresponding to crop polygons are assigned a value of 1, while pixels corresponding to non-crop polygons are assigned a value of 0. The resulting non-overlapping areas and cloud pixels are assigned to class 2 as unknown. The resulting mask therefore defines the spatial extent of valid labeled data, distinguishing between crop, non-crop and unknown regions.

#### 4.5.8.2 Boundary Layer

For the Boundary Layer, the crop boundaries are obtained from the geometric boundaries of the crop polygons in the in-situ data. Thereafter, non-crop polygons are inwardly buffered by one pixel before rasterization, effectively shrinking their extent and creating a margin near crop boundaries to reduce potential edge ambiguity and ensure separation between classes. The Boundary Layer is then constructed such that boundary pixels are assigned a value of 1, while both crop interiors and buffered non-crop areas are assigned a value of 0. This mask therefore isolates the field boundaries while suppressing surrounding regions.

#### 4.5.8.3 Distance Layer

In order to compute the Distance Layer, a Euclidean distance transform is applied to the rasterized Boundary Layer, with distances computed for all non-boundary pixels to the nearest boundary pixel. The resulting distance values represent the shortest straight-line distance to a field edge for each pixel. Distances are then constrained to only crop areas and non-crop

areas are explicitly assigned to 0. Normalization is done using a maximum distance which is the same one used in the pre-training. This results in a representation of proximity to field boundaries within crop regions.

#### 4.5.8.4 Display of Training Inputs

A subset of each training input layer is displayed in Figure 20.

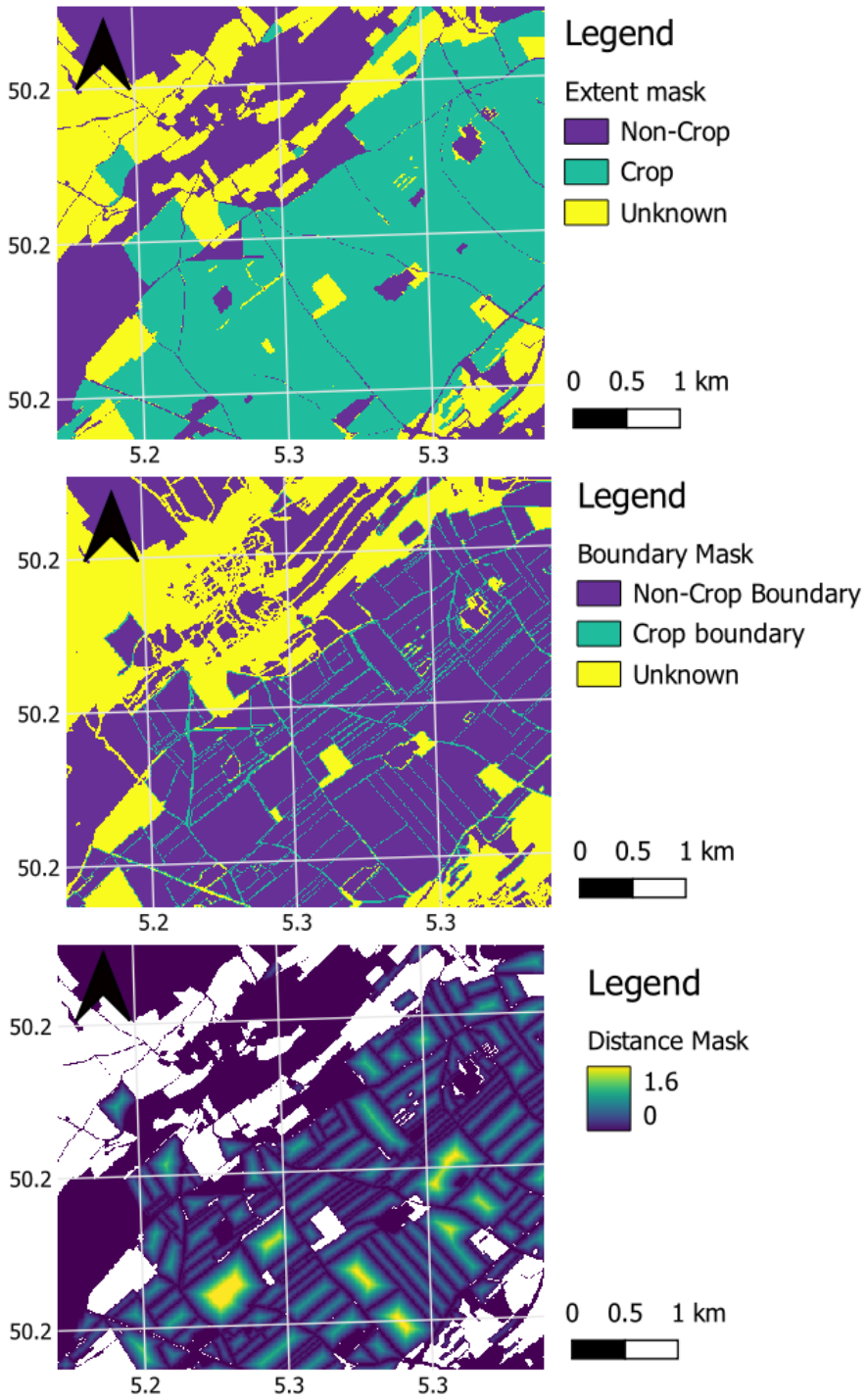



Figure 20: Extent, Boundary and Distance Inputs for Training for 100% Dataset.

### 4.5.9 Making the Training Dataset

Given that a large amount of data composes the training dataset, the dataset is subdivided into smaller patches to make the training process more computationally manageable. In this case, a patch is defined as an area of 256 by 256 pixels.

A sliding window approach is applied across the entire extent mask, where patch coordinates are extracted at regular intervals corresponding to the patch size. To avoid edge effects during model training, a small contextual margin is considered, ensuring that only patches fully contained within the valid raster extent are used. Furthermore, the patches that fall anywhere within the project's validation areas, are considered to be validation patches, and the remaining patches within the training zones are training patches.

At this stage, a cloud filtering step is applied to reduce the presence of cloudy pixels in both the training and validation datasets. If inside a patch, for any given month the cloud cover exceeds 20% of the pixels, then that patch is excluded from the training and/or validation. The included patches are valid patches for further analysis. This cloud filtering step mainly affects TA1, where out of 120 training patches and 18 validation patches, only 39 and 12 training and validation patches remain respectively. For TA2, two training patches are excluded and TA3 is not affected.

Following the extraction of patches into training and validation sets, the labels are extracted from the previously created Extent, Boundary, and Distance layers. For each patch, these three values are extracted per pixel, and combined into a three-channel label layer. These label patches are then stored for subsequent use in model training and validation. The extracted label patches have a dimension of (256, 256, 3), where 256 is the original patch size in both the x and y direction, and 3 is the number of labels extracted.

Subsequently, the features are extracted from the monthly and bi-monthly NDVI composites. For each training area, the NDVI composites corresponding to the selected months are stacked into a multi-band array, where each band represents a different month.

Each band is normalized independently using percentile-based scaling (1st–99th percentile), after which values are constrained to the range between 0 and 1. This normalization reduces the influence of outliers and ensures that all temporal inputs contribute comparably to the model, preventing bias toward specific months with larger or lower value ranges. The min-max statistics used for normalization are subsequently saved to be used later in the prediction step.

The feature extraction patches have a size of (256 + 3*2, 256 + 3*2, 6), where 256 is the original patch size in both the x and y directions, and 3 represents the context size added on each side (2) of the patch. Therefore, the dimensions become 256+6 = 262 pixels in both spatial directions, resulting in a feature extraction size of (262, 262, 6). The role of the spatial context added is to extend the extraction window beyond the patch boundaries to incorporate neighbouring information. This is done so that when the CNN trains, pixels near patch edges have a comparably similar amount of context as interior patch pixels. Finally, the third dimension (6), is the number of features extracted, one for each NDVI composite. Using the previously defined patch coordinates, feature patches are then extracted from the normalized NDVI stack. The resulting feature patches are stored for both training and validation sets.

### 4.5.10 Model Training and Architecture

Model training is performed using a CNN2D model. As previously explained, for the Scratch model, training is performed individually for each of the in-situ datasets (100%, 50%, 25%, 10%, 5%, 4%, 3%, 2%, 1%). Meanwhile, for the Pretrained model, first pre-training is first performed, then training is performed individually for each of the in-situ datasets.

Figure 21 below depicts the architecture of the CNN2D model used.

The CNN2D model takes as input image patches of size 262×262 pixels for each of the 6 NDVI monthly composite inputs used. The network begins with a convolutional layer of 128 filters with a 7×7 kernel, which extracts low-level spatial features from the input patches. Activation functions including ReLU and Sigmoid, are applied throughout various steps in the CNN to introduce non-linearity into the network. Batch normalization is applied to improve training stability.

Two residual blocks are used to progressively reduce the feature dimensionality while preserving spatial context through skip connections. Each residual block consists of a main path, where convolutional layers learn feature transformations, and a skip connection, which directly forwards the input feature maps to the output of the block. The outputs of the two paths are combined in the Add layer using an element-wise addition operation.

The second residual block produces the final feature maps, which are then passed through separate 1×1 convolutional layers to create the three output feature map predictions of: Extent, Boundary and Distance. In the Training Supervision, a Tanimoto loss function that ignores unknown labels is used to compute the error between predicted and label Extent, Boundary, and Distance maps. The model is trained using the Adam optimizer with a learning rate of 0.0001, batch size of 16, early stopping patience of 5 epochs, and a Tanimoto loss function.

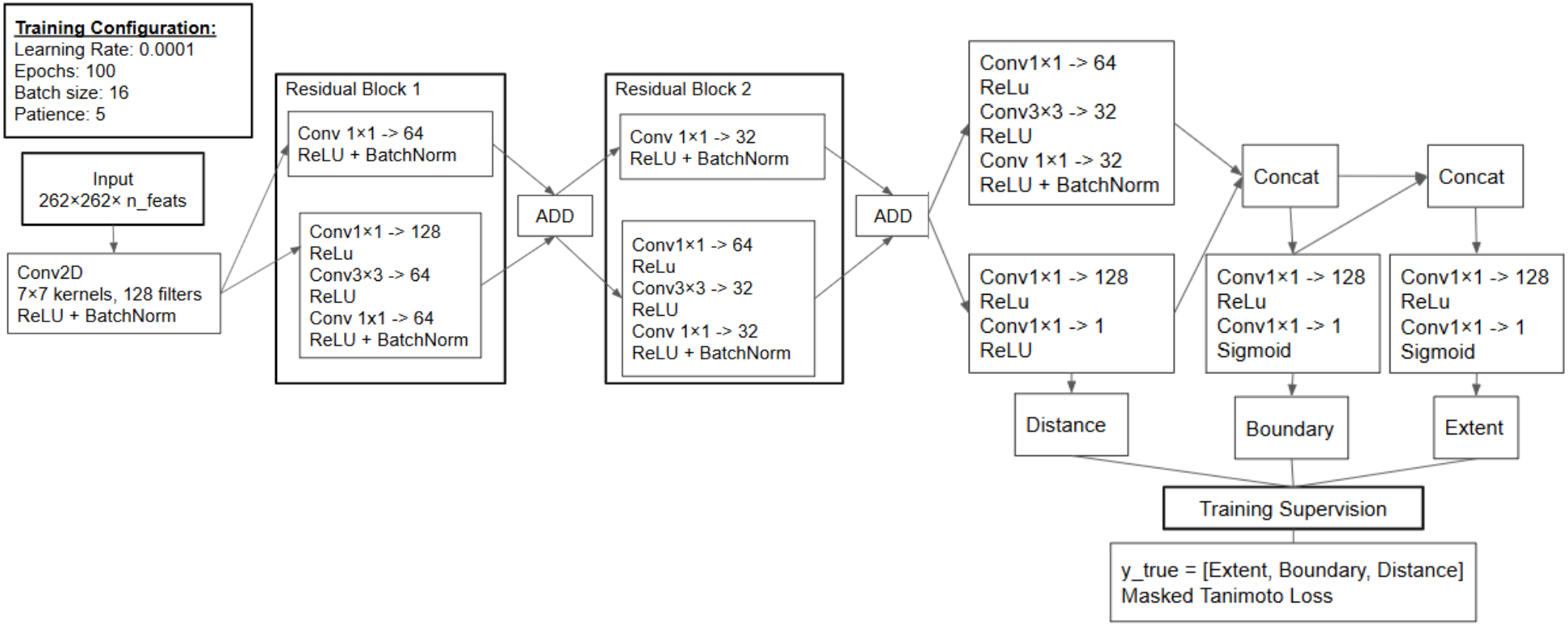



Figure 21: Architecture and Hyperparameters Setting of the CNN2D Model.

### 4.5.11 Prediction

Model prediction is performed over the Wallonia AOI using the previously saved trained models. First, the same monthly and bi-monthly NDVI median composites used for training but covering the entire prediction AOI are used in the same order as for training. These are individually normalized using the precomputed min–max statistics derived from the feature extraction. The study area is thereafter divided into non-overlapping patches of 256 by 256 pixels, with an additional spatial context margin applied around each patch to preserve local spatial continuity at patch boundaries.

The trained CNN model is then applied to all extracted patches to make a prediction. Each patch produces three output maps corresponding to pixel probability of crop Extent, Boundary, and Distance to boundary. Finally, the resulting multi-channel prediction is made to preserve the original spatial reference and resolution.

After model prediction, the three output layers (Extent, Boundary, and Distance) are processed to derive vectorized field boundaries. First, the extent layer is used with a threshold to retain only pixels likely belonging to agricultural areas. The Distance layer is smoothed using a Gaussian filter and combined with the boundary prediction, where boundary values help define separations between fields, while distance values help delineate individual field regions.

A watershed segmentation is subsequently applied, where local maxima in the distance layer serve as seeds for individual fields. This results in a labeled raster in which each region corresponds to a distinct field polygon. The labeled regions are then vectorized into polygons yielding a vectorized set of field boundary predictions.

### 4.5.12 Model evaluation

The models that are run for each percentage of in-situ data are each evaluated by comparing the predicted field boundaries with the LPIS dataset only including agricultural field polygons for all of Wallonia. The field boundary predictions are compared on a pixel by pixel basis with the declared LPIS agriculture boundaries and evaluation metrics are computed.

The MCC, which is a pixel based metric, is computed to evaluate how well individual pixels are correctly or incorrectly classified including TP, TN, FP and FN values. The IoU metric is calculated using polygon geometries and by retaining the highest IoU values among intersection polygons. Then the IoU is expressed in two ways: the IoU ≥ 0.5 and the median IoU. The IoU ≥ 0.5 metric represents the proportion of reference polygons with an IoU value greater than or equal to 0.5, thereby measuring how many predicted polygons achieve the set minimum acceptable overlap threshold. In contrast, the median IoU represents the median overlap accuracy across all matched polygons and provides an indication of the overall segmentation quality. While the median IoU evaluates the typical overlap performance across all polygons, the IoU ≥ 0.5 metric focuses on the frequency of sufficiently accurate detections. Together, the MCC and IoU metrics provide a quantitative assessment of how well the predicted field boundaries match the reference dataset.

# Chapter 5

# Results

The obtained results are explained in this section in terms of training, prediction and model performance.

## 5.1 Training

During model training, the training and validation loss are tracked in order to determine when to stop model training. The training loss across the epochs used to train each model is shown in Figure 22 below. As the percentage of in-situ data used increases, the starting training loss decreases, as the model has more data to learn relationships in the data. Additionally, the Pretrained models all exhibit lower training losses compared to corresponding Scratch models throughout the whole training. Moreover, the Scratch models exhibit higher training loss in early epochs which rapidly decreases, but does not reach as low levels as the corresponding Pretrained model.

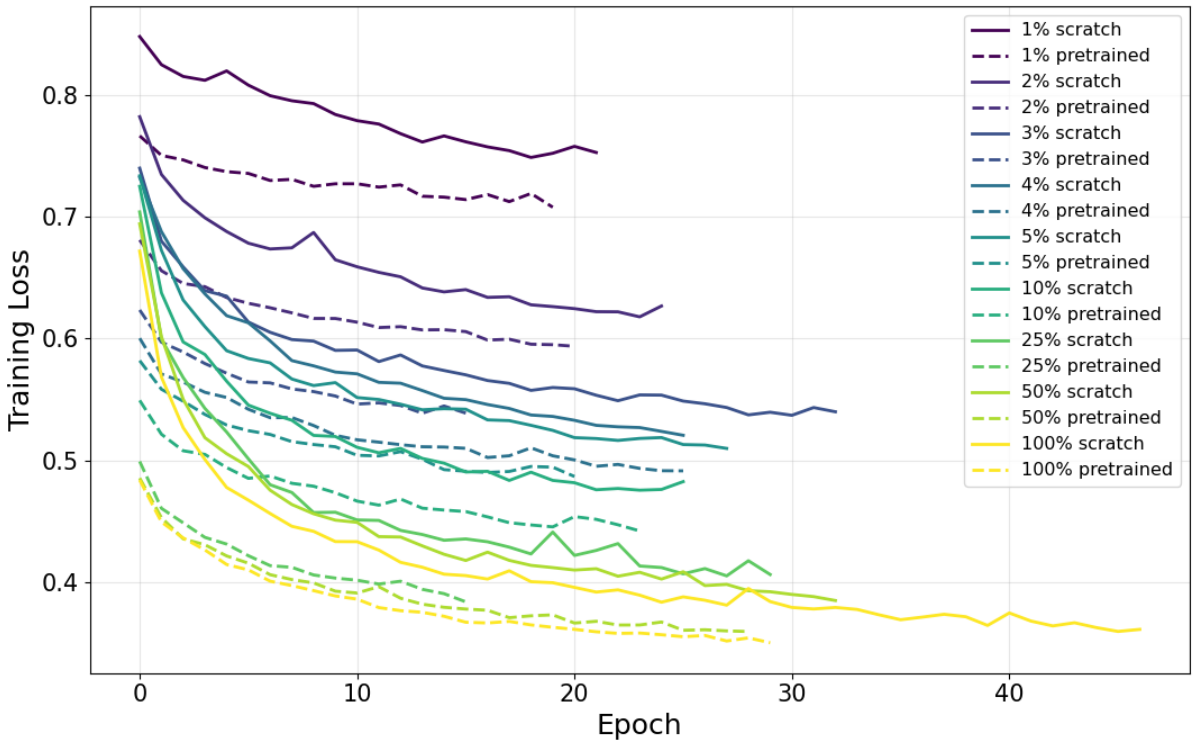



Figure 22: Training Loss Across Epochs for Scratch and Pretrained Models over Training Areas.

In Figure 23, the validation loss across the epochs used to train each model is displayed. The patterns found resemble those of the training loss, with the exception that the validation loss generally exhibits greater variability and stronger fluctuations between epochs. This occurs as the data used for validation has previously not been seen by the model.

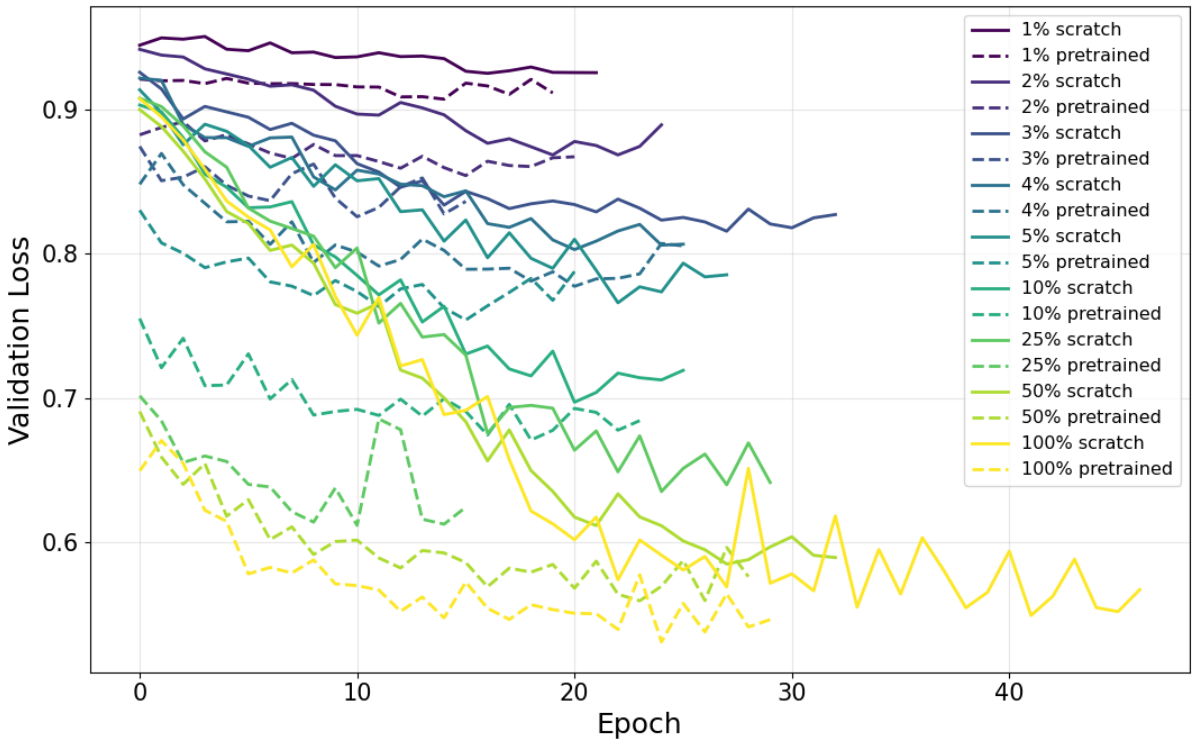



Figure 23: Validation Loss Across Epochs for Scratch and Pretrained Models over Validation Areas.

In addition, the number of epochs is lower for the Pretrained models than the corresponding Scratch model runs for all runs, except for 4% where the number of epochs is the same. In Table 3, the number of epochs used for each model is shown along with the percent difference between the Scratch and Pretrained corresponding models. The number of epochs decreases from 0% to -52% for the Pretrained models, with an average percent difference of -23%.

Table 4: Number of Epochs used for Model Training for each in-situ data Percentage

| Number of Epochs | 100% | 50% | 25% | 10% | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|---|---|---|---|
| Scratch | 47 | 33 | 30 | 26 | 28 | 26 | 33 | 25 | 22 |
| Pretrained | 30 | 29 | 16 | 24 | 21 | 26 | 16 | 21 | 20 |
| Percent difference | -36% | -12% | -47% | -8% | -25% | 0% | -52% | -16% | -9% |

## 5.2 Prediction

Each model produces a prediction of field boundaries over all of Wallonia, from which it is possible to calculate the predicted agricultural area and number of agricultural fields. These values for each model can be found in plots a) and b) of Figure 24.
The total area of the agricultural fields in the LPIS dataset is 4,391.5 km². In terms of predicted agricultural area, both the Scratch and Pretrained models substantially overestimate the area with predictions exceeding more than twice the actual extent. The Scratch model shows a

stronger tendency toward overestimation compared to the Pretrained model. As the percentage of in-situ data decreases, estimated agricultural area steadily increases. For the 1% Scratch model, the agricultural area greatly increases and reaches 15,744 km², while for the Pretrained model it remains at 8,154 km².

In regards to the number of agriculture fields, in the LPIS for Wallonia, there exist 161,740 agriculture fields. The Scratch models under or over estimate the number of fields by about 10,000 for percentages 50% to 3%, with a more extreme underestimation for the 100% run. For percentages 2% to 1%, the predicted number of fields plummets greatly. The Pretrained model underestimated the number of agriculture fields by an average of all percentages of 19.2%.

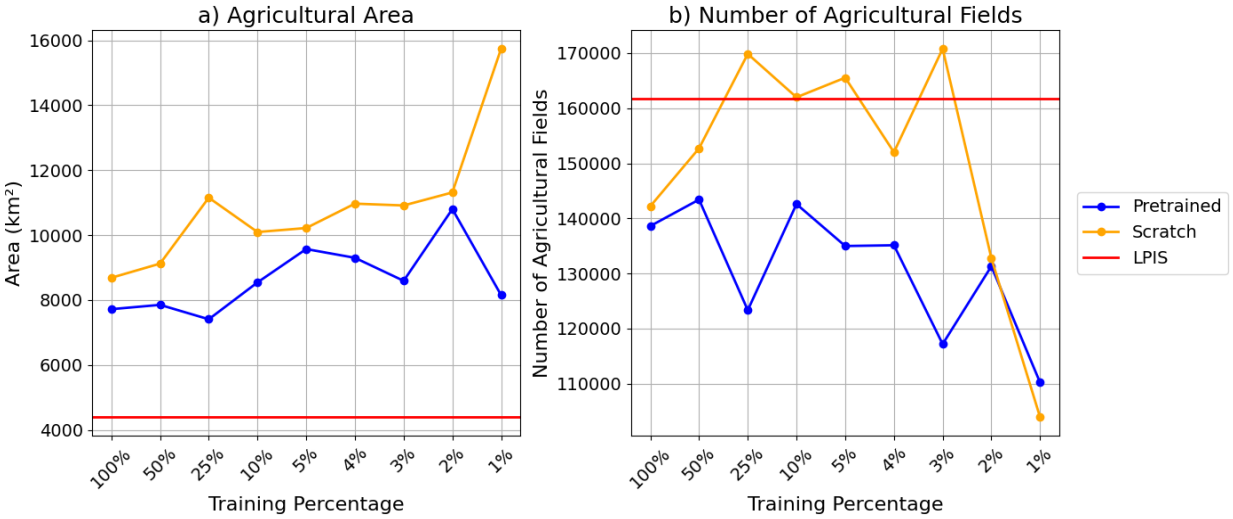



Figure 24: Predicted and actual agricultural area (a) and number of agriculture fields (b) in Wallonia.

Furthermore, the model predictions and vectorized outputs of field boundaries are displayed for a Subset area within Wallonia to aid interpretability. The chosen area, which is not within any of the model training areas, is the one shown in Figure 25 with the LPIS field boundaries.

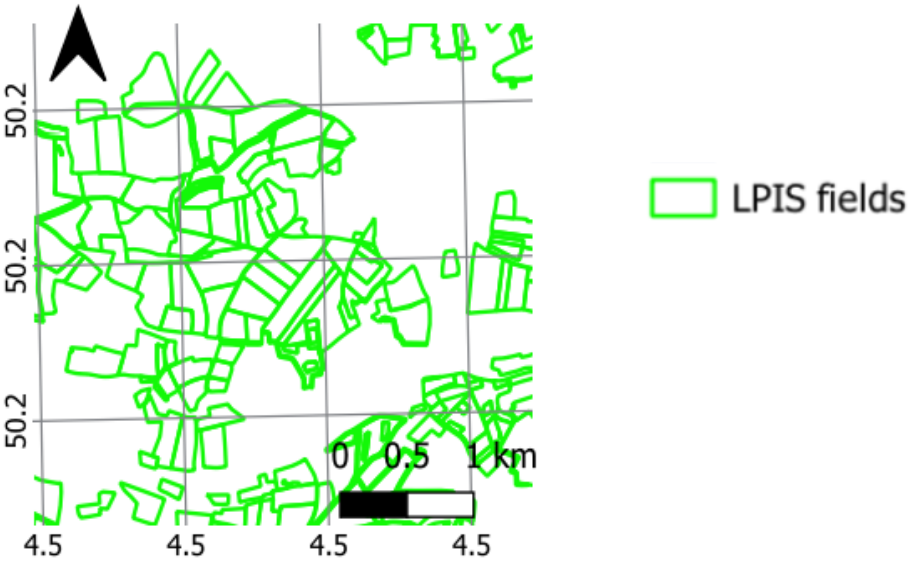



Figure 25: LPIS fields in Subset in area not Included in Model Training.

In Figure 26 below, the predicted and vectorized model outputs are found for the 100% in-situ dataset within the subset in area for the Scratch and Pretrained models. Similarly, in Figure 27, the same is shown for the 1% dataset. Both figures include a color composite of the three band model prediction, the model prediction with the predicted fields, the model prediction with the LPIS fields, the LPIS fields on top of the predicted fields and finally the predicted fields on top of the LPIS fields.

To aid the interpretability of the prediction color composites in Figure 26 (a-c) and Figure 27 (a-c) a detailed ternary RGB legend is found in Appendix A. In the making of the color composites, the Extent, Boundary, and Distance prediction bands are assigned the red, green, and blue color channels, respectively. The predicted probability values are represented using a continuous RGB color space derived from the combination of these three channels. Consequently, high Extent and Boundary probabilities are depicted in yellow, cyan represents high Boundary and Distance and magenta high Extent and Distance probabilities. Pixels with low probabilities across all three prediction bands appear in black, whereas pixels with high probabilities in all three bands are displayed as white.

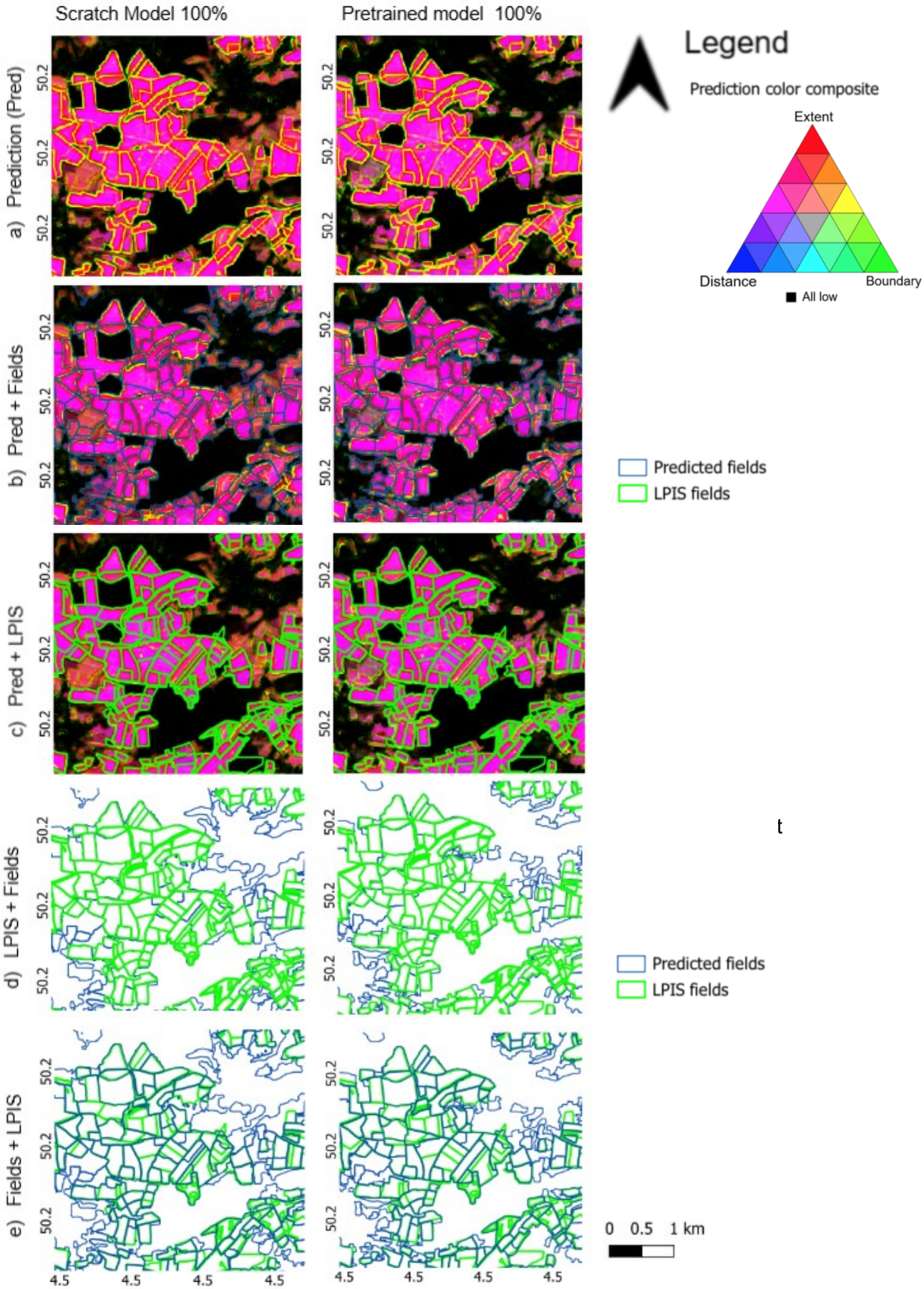



Figure 26: Prediction and Vectorized Model Outputs for 100% in-situ data in Subset area.

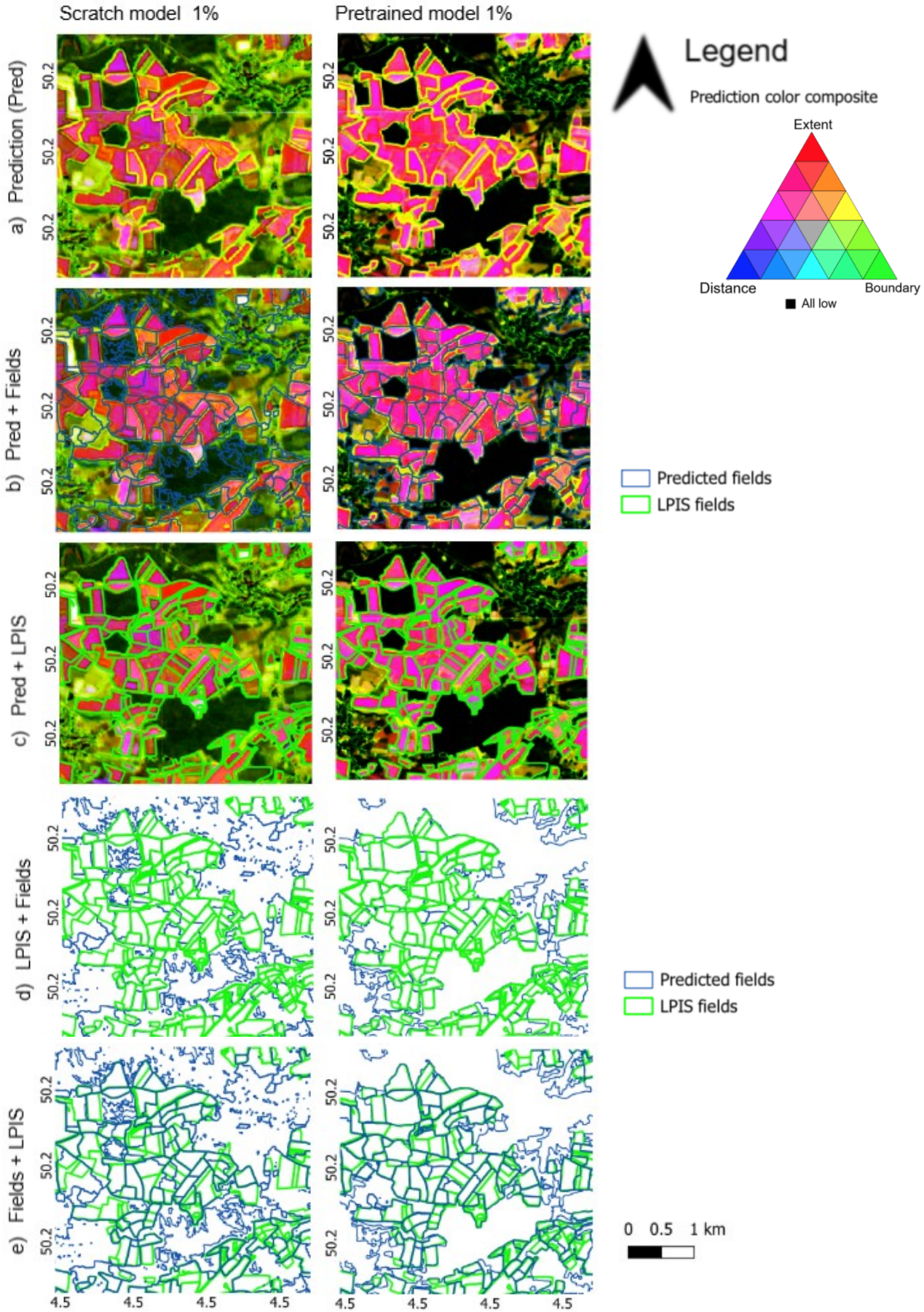



Figure 27: Prediction and Vectorized Model Outputs for 1% in-situ data on Subset area.

In terms of the model predictions shown in Figures 26 and 27 in a), b), and c) bright magenta or pink tones indicate areas with strong extent and distance values, while green or yellow edges emphasize detected field boundaries. From the visual interpretation, the prediction results of the Scratch and Pretrained models for the 100% dataset are largely comparable in Figure 26. Agreement between the two models is particularly evident in regions characterized by high-confidence predictions, represented by bright pink areas with clearly defined yellow boundaries. In contrast, differences between the models are more apparent in regions with lower prediction confidence, represented by less bright and more spatially diffuse pink areas. Furthermore, Figure 26 b) indicates that many of these uncertain regions are vectorized into field polygons by both models, whereas comparison with Figure 26 c) shows LPIS fields are not present there. Additionally, Figures 26 d) and e) demonstrate that correctly classified field boundary pixels are generally similar between the two models, while greater differences in segmentation performance are observed in non-agricultural areas.

When comparing the predictions generated using the 1% dataset with those from the 100% dataset, the most pronounced difference in Figure 27 a) is observed in the Scratch model predictions. In comparison with Figure 26 a), the Scratch model in Figure 27 a) exhibits substantially greater variation in hue and intensity within the pink regions, indicating less consistency within field areas. Additionally, regions that appear predominantly black in Figure 26 a) are instead characterized by increased green lines and an overall green hue Figure 27 a).

The 1% Pretrained model prediction more closely resembles the 100% predictions, but with more green tones in the non-agricultural areas. Furthermore, the general idea of the field polygons is obtained in the 1% dataset by the Scratch and Pretrained models, shown by the vectorized fields in e) matching the LPIS fields in d). In contrast, the Scratch 1% model demonstrates substantially greater segmentation errors in non-agricultural areas in Figure 27 e), where extensive regions are incorrectly classified as single large agricultural fields containing numerous smaller fragmented polygons. This suggests that the model takes the outside boundaries of fields and merges them, failing to adequately distinguish between separate field and non-field regions.

In the western part of Wallonia, the density of agricultural fields is lower compared to the remainder of the study area, which leads to a distinct spatial prediction pattern in this region. This is illustrated in Figure 28, where both Scratch and Pre-trained models show many predicted fields in areas where no fields are present in the LPIS dataset.

The Pretrained 100% and Pretrained 1%, show very similar segmentations. There is slightly more density of fields in the Scratch 100% compared to the Pretrained 100%. Finally, the Scratch 1% exhibits some incorrect segmentations of very large fields.

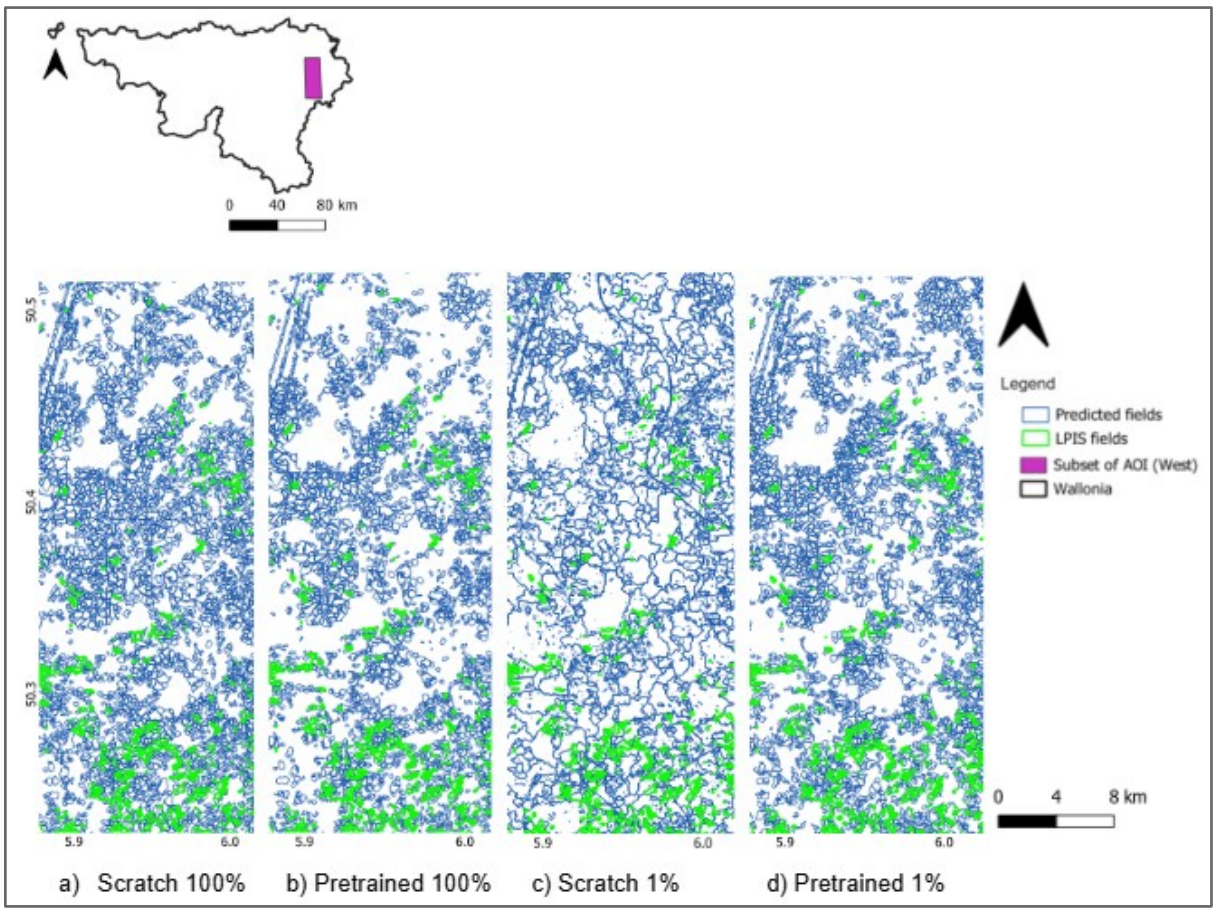



Figure 28: Vectorized field predictions in Subset of AOI (West) for Scratch and Pretrained Models of 100% and 1% in-situ Datasets.

This Western area of Wallonia is largely composed of pastures and meadows. Pastures and meadows are a land use class from the LPIS excluded from training. The area outside of the LPIS agricultural fields can therefore be considered as the area of possible incorrect model predictions. Within this potential misclassification area, 25% of it consists of pastures and meadows, 69% non-crop areas, 6% unknown areas and the other land classes excluded from the LPIS dataset which compose less than 1% of the area combined. However, on average across all dataset percentages, 74% of these false predictions occur on pastures and meadows, while the remaining 26% are distributed across the other land-use classes. The mean and standard deviation of predictions on incorrect land types are shown in Table 5, while

the percentage of area per incorrect land type for each dataset percentage are available in Appendix C.

Table 5: Average and Standard Deviation of Field Prediction Area in Incorrectly Predicted Land Types

| Land type | Field Area Average | Field Area Standard Deviation |
|---|---|---|
| Pastures and Meadows | 74% | 4% |
| Others | 26% | 4% |

## 5.3 Model Performance

The vectorized model predictions are evaluated in terms of the entire Wallonia AOI as well as exclusively for agriculture fields within the Wallonia AOI.

### 5.3.1 Model Performance on full Wallonia AOI

The evaluation of all models on a pixel level with the MCC metric is presented in Table 6, and is visually represented in Figure 29.

The MCC ranges from 0.63 to 0.41 for the Scratch models at 100% and 1% respectively, and from 0.69 to 0.55 for the Pretrained models at 25% and 2%. For both types of models, the MCC generally decreases as the percentage of in-situ data decreases with some exceptions. For the Scratch model, there is a notable drop in MCC from the 2% model to the 1% model, while the MCC at 1% of the Pretrained model is 0.65. The Pretrained model achieves higher MCC scores than the Scratch model for all corresponding in-situ data percentages.

Table 6: MCC for Pretrained and Scratch Models Along in-situ data Percentages

| MCC | 100% | 50% | 25% | 10% | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|---|---|---|---|
| Scratch | 0.63 | 0.62 | 0.54 | 0.58 | 0.58 | 0.55 | 0.55 | 0.54 | 0.41 |
| Pretrained | 0.68 | 0.68 | 0.69 | 0.64 | 0.60 | 0.61 | 0.64 | 0.55 | 0.65 |

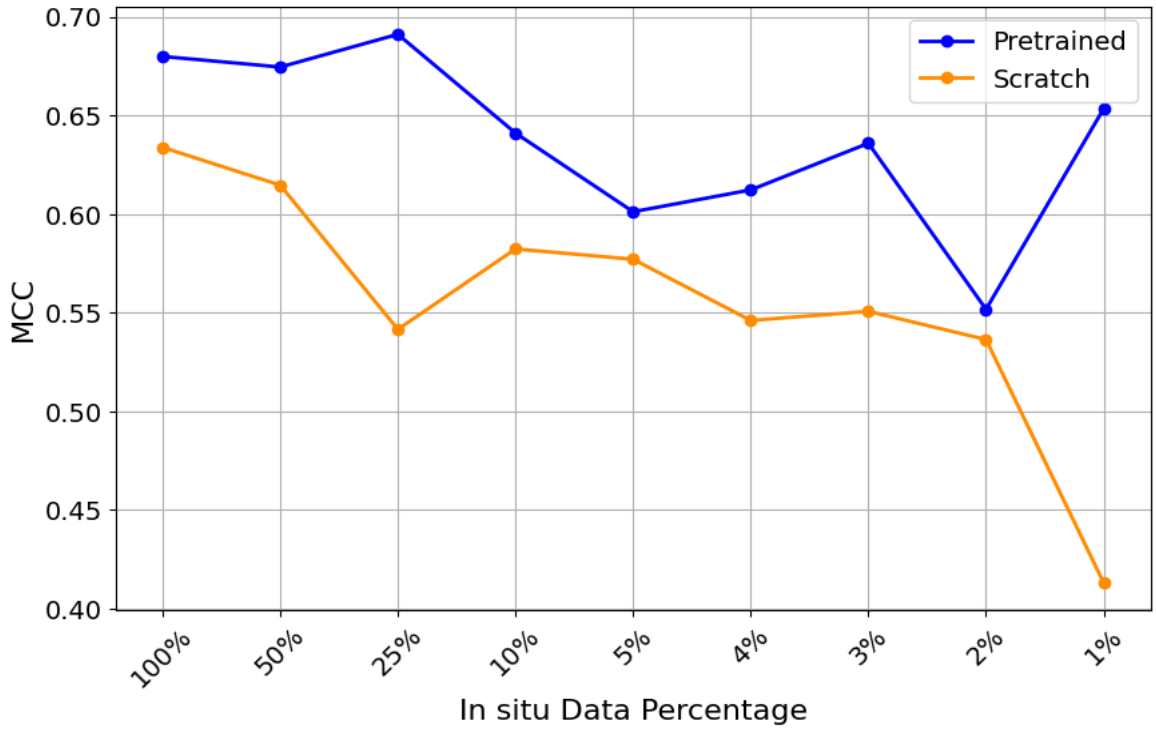



Figure 29: MCC for Pretrained and Scratch Models Along in-situ data Percentages.

The IoU object based evaluation metric results are shown in Table 7 and Figure 30 including the median IoU and IoU ≥ 0.5. Regarding the Scratch model, the range for the IoU ≥ 0.5 ranges from 0.31 at 100% and reaches 0.16 at 1%, while the median IoU remains at a constant 0.24 from 100% to 25% and reaches 0.11 at 1%. In terms of the Pretrained model, the range for the IoU ≥ 0.5 is from 0.32 at 100% to 0.23 at 1%, and the median IoU obtains values from 0.26 at 100% and 0.18 at 1%.

Both IoU metrics decrease gradually and marginally as percentage of training data decreases, with no major difference in range between the Scratch and Pretrained models. However, for the Scratch model, both the IoU ≥ 0.5 and median IoU exhibit a considerable drop from 2% to 1%, with IoU ≥ 0.5 decreasing to 0.16 and median IoU to 0.11 at 1% training data. This substantial decrease is not observed for the Pretrained 1% model, where the IoU ≥ 0.5 is 0.23 and median IoU is 0.18.

Table 7: IoU for Pretrained and Scratch Models Along in-situ data Percentages

| Model | Metric | 100% | 50% | 25% | 10% | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|---|---|---|---|---|
| Scratch | IoU ≥ 0.5 | 0.31 | 0.31 | 0.30 | 0.29 | 0.29 | 0.27 | 0.28 | 0.25 | 0.16 |
| | Median IoU | 0.24 | 0.24 | 0.24 | 0.24 | 0.24 | 0.22 | 0.23 | 0.20 | 0.11 |
| Pretrained | IoU ≥ 0.5 | 0.32 | 0.31 | 0.29 | 0.30 | 0.27 | 0.28 | 0.25 | 0.26 | 0.23 |
| | Median IoU | 0.26 | 0.25 | 0.23 | 0.24 | 0.22 | 0.22 | 0.20 | 0.20 | 0.18 |

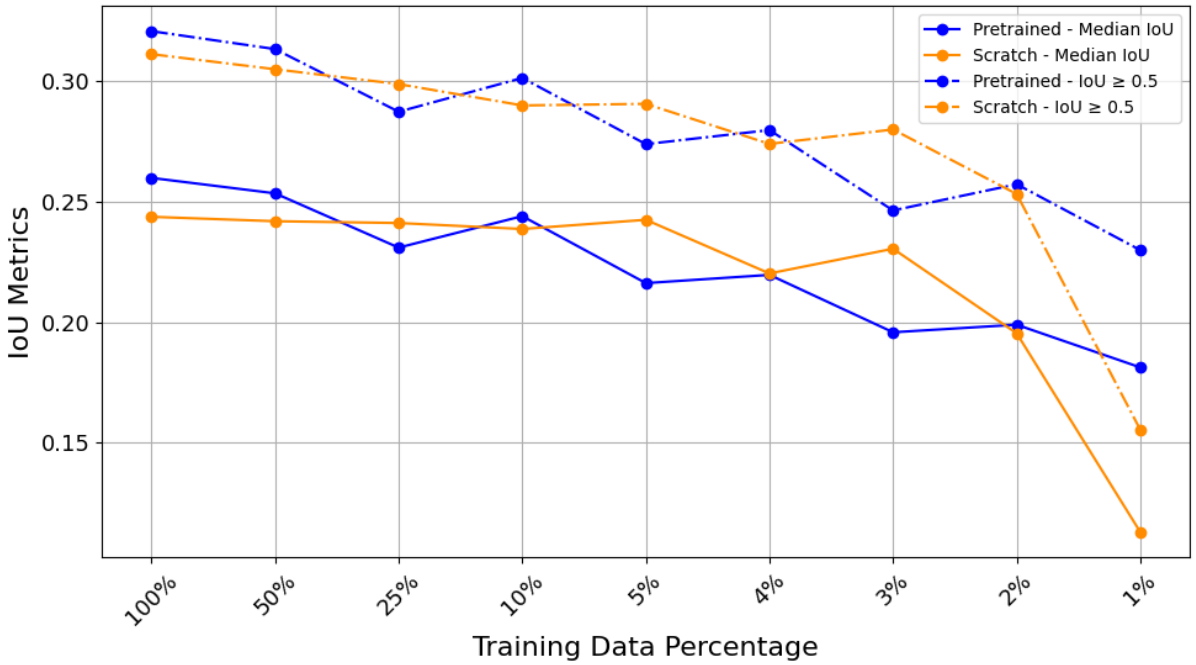



Figure 30: IoU for Pretrained and Scratch Models Along in-situ data Percentages.

### 5.3.2 Model Performance on agriculture field areas

Given the overprediction of agricultural field boundaries in non-crop areas, the model predictions are further evaluated only in agricultural areas. This is done to determine whether a more representative assessment of the model's field delineation capability can be obtained by excluding a known source of error that disproportionately affects the overall performance metrics. The vectorized predictions are clipped to the extents of agricultural field polygons in the LPIS dataset and evaluated. The MCC values for this analysis are in Table 8 and visually represented in Figure 31.

As can be seen, MCC values range from 0.98 to 0.99 for all model runs including the Scratch and Pretrained models. This indicates near perfect agreement between predicted and reference field boundaries.

Table 8: MCC for Pretrained and Scratch Models Along in-situ data Percentages in Agricultural Field Area

| MCC | 100% | 50% | 25% | 10% | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|---|---|---|---|
| Scratch | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| Pretrained | 0.98 | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 | 0.98 | 0.99 | 0.98 |

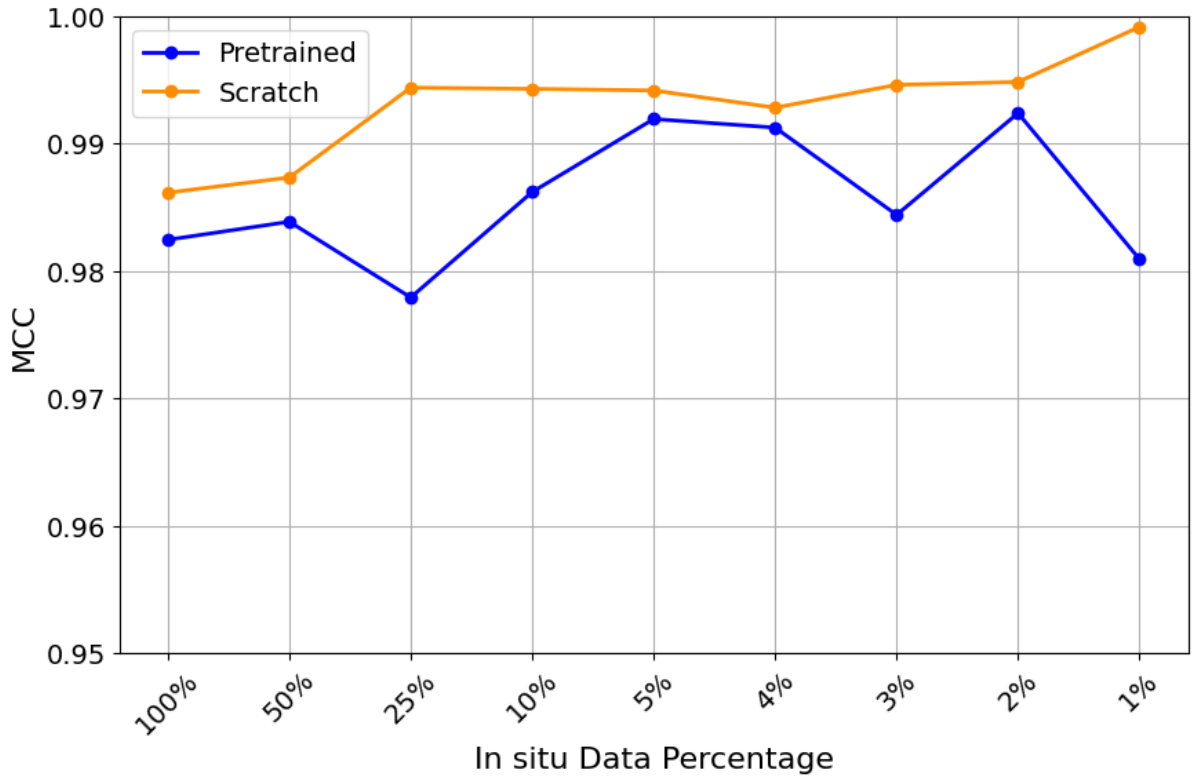



Figure 31: MCC for Pretrained and Scratch Models Along in-situ data Percentages in Agricultural Field Area.

The IoU evaluation results are displayed in Table 9 and Figure 32.

For the Scratch models, the IoU ≥ 0.5 decreases from 0.41 to 0.24 as the amount of training data is reduced, whereas the corresponding values for the Pretrained models range from 0.42 to 0.33. The median IoU which declines from 0.39 to 0.21 for the Scratch models and from 0.38 to 0.28 for the Pretrained models. For both models, the reduction in performance is relatively gradual between the 100% and 5% model runs, followed by a more pronounced decline at the lower percentages.

Compared to the evaluation in the complete Wallonia AOI, the IoU metrics in agriculture areas are consistently higher. Additionally, when comparing Figure 32 to Figure 30, the shape of the plot lines are conserved along the percentage runs. Furthermore, the overall trends observed across the different training data percentages remain largely unchanged. As illustrated by the comparison between Figure 32 and Figure 30, the shape of the performance curves is preserved.

Table 9: IoU for Pretrained and Scratch Models Along in-situ data Percentages in Agricultural Field Area

| Model | Metric | 100% | 50% | 25% | 10% | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|---|---|---|---|---|
| Scratch | IoU ≥ 0.5 | 0.42 | 0.41 | 0.42 | 0.41 | 0.41 | 0.39 | 0.40 | 0.37 | 0.24 |
| | Median IoU | 0.38 | 0.37 | 0.39 | 0.38 | 0.38 | 0.36 | 0.38 | 0.34 | 0.21 |
| Pretrained | IoU ≥ 0.5 | 0.42 | 0.41 | 0.39 | 0.41 | 0.39 | 0.39 | 0.35 | 0.37 | 0.33 |
| | Median IoU | 0.38 | 0.37 | 0.33 | 0.37 | 0.35 | 0.35 | 0.30 | 0.33 | 0.28 |

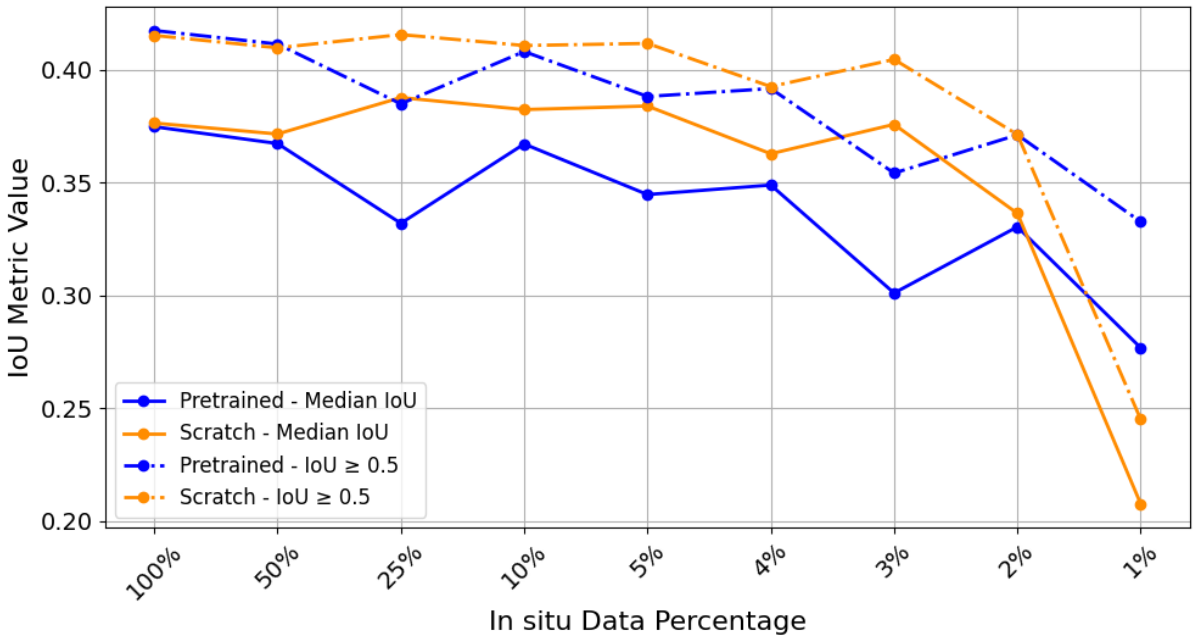



Figure 32: IoU for Pretrained and Scratch Models Along in-situ data Percentages in Agricultural Field Area.

# Chapter 6. Case Study: Applying Wallonia methodology on Pakistan AOI to observe pre-training effect on boundary delineation.

This Case Study applies the present study's methodology for pre-training a CNN using EL to determine the transferability of the methodology in another area. As Pakistan is an agricultural in-situ data poor region it is an interesting region to perform field boundary delineation using limited amounts of in-situ data. The first step is to determine if the methodology used for Wallonia is applicable for Pakistan or if it must be previously adapted. The purpose is to observe if Pakistan outputs are similar to those of Wallonia and if pre-training has an effect on the data. The evaluation of predictions is not the most important aspect of this Case Study.

**Objective**: To compare the effects of pre-training a CNN in Pakistan for field boundary delineation using the same methodology as for Wallonia using a 100% in-situ dataset.

**Area of Interest**:
The AOI is located in southern Pakistan and is composed of two Training Areas (TAs) and one Validation Area (VA) shown in Figure 31. The Prediction Area (PA), which is the region the model will predict, is on the two TA and one VA areas used for training and validation due to lack of data. A comparison between the Wallonia and Pakistan AOIs is in Table 8.

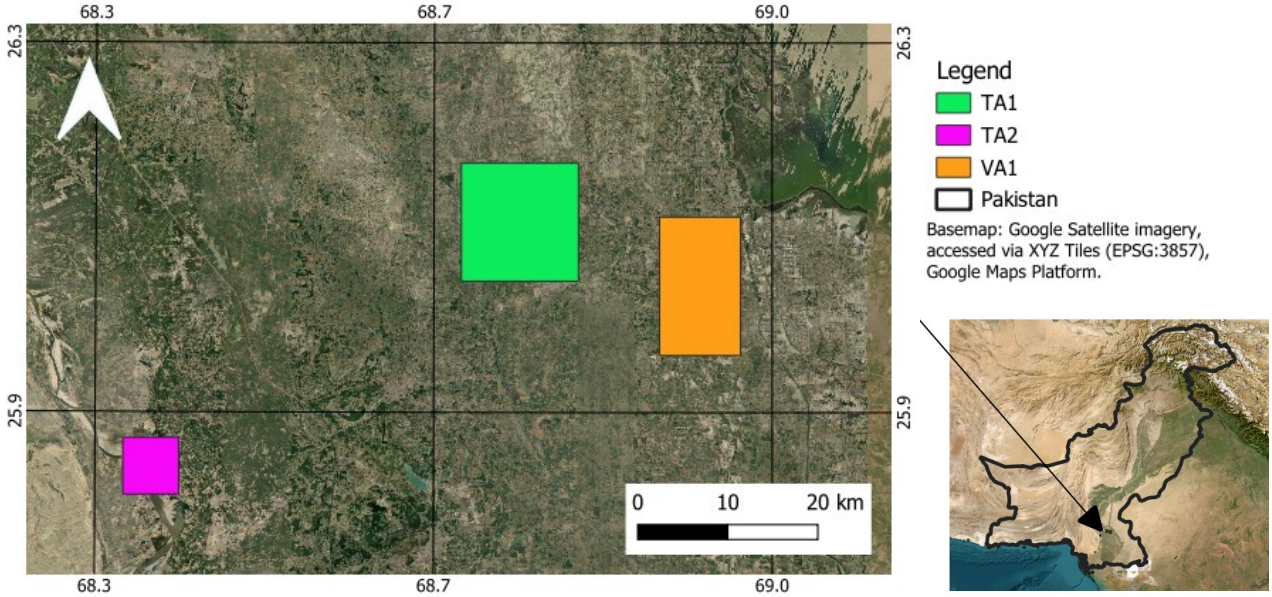



Figure 33: Training and Validation areas in Pakistan AOI.

Table 10: Area and Number of crop and non-crop fields for Pakistan and Wallonia AOIs *

| AOI | Total Area (km2) | Training Area (km2) | Validation Area (km2) | Pre-training Area (km2) | Total crop fields in TAs | Total non-crop polygons in TAs |
|---|---|---|---|---|---|---|
| Pakistan | 347.9 | 209.6 | 138.3 | 347.9 | 185,601 | 163,561 |
| Wallonia | 16,988.6 | 3,720.8 | 425.8 | 16,988.6 | 47,021 | 259,455 |

* values from before patch creation and before cloud filtering

**Data used:**
In situ data used: The in-situ data comes from a segmentation of crop and non-crop boundaries performed by the UCLouvain Geomatics Research Lab. Only a 100% in-situ data set is used.
RS data: Sentinel-2 for NDVI composites

Patch size: 256 x 256 pixels
Training patches: 29
Validation patches: 15
Pre-training patches: 44

**Model used:** Same CNN2D as for Wallonia AOI.

**Months used:** 10-2023, 11-2023, 12-2023, 01-2024, 02-2024, 03-2024
Due to high cloud cover, months 09-2023, 01-2024, 02-2024 are excluded.

**Methodology:**
Scratch model Training:
For TA and VA areas:

1. From NDVI median monthly composites obtain Features
2. From in-situ data create Labels:
   - Extent
   - Boundary
   - Distance
3. Extract Features and Labels to create Training Dataset
4. Train model
5. Predict on PA.
6. Vectorize predictions.
7. Evaluate vectorized predictions on PA with in-situ data.

PRETRAINED model Pre-training:
For TA and VA areas:

1. From NDVI median monthly composites obtain Features.
2. From in-situ data create pseudo-Labels:
   - Extent: From in-situ data
   - Boundary: UL Classifiers: 22 used, same as in Table 3 for Wallonia AOI.
   - Distance : Calculated from Boundary Layer
3. Extract Features and pseudo-Labels to create Training Dataset
4. Train for 1 epoch and save model.

PRETRAINED model Training:

5. From in-situ data create Labels:
   - Extent
   - Boundary
   - Distance
6. Extract Features and Labels to create Training Dataset
7. Train model using Pretrained weights for initialization.
8. Predict on PA.
9. Vectorize predictions.
10. Evaluate vectorized predictions on PA with in-situ data.

**Training Results and Interpretation:**


Figure 34: Training loss of models over epochs in training areas

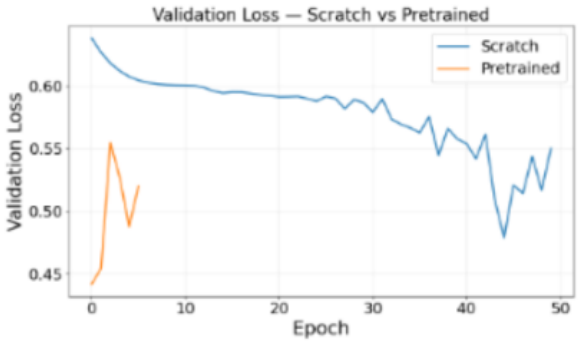



Figure 35: Validation loss of models over epochs in training areas

· Scratch model takes 50 epochs to train, while Pretrained model takes 6 (Table 9).
· 88% reduction in training epochs from the Pretrained model compared to Scratch model.
· Training loss starts lower for the Pretrained model (0.63 Pretrained vs 0.71 Scratch).
· Training loss finalizes lower for Scratch model (0.42 Scratch vs 0.46 Pretrained).
· Compared to Wallonia study:
  · Pakistan case study has the lowest number of epochs for the Pretrained model and the highest number for the Scratch model than any other run.
  · Final training and validation loss in Pakistan is similar to Wallonia 100% models.

Table 11: Training and Validation Results for Scratch and Pretrained model runs

| Model | Number of Epochs | Training loss Epoch 1 | Training loss Final Epoch | Validation loss Epoch 1 | Validation loss Final Epoch |
|---|---|---|---|---|---|
| Scratch | 50 | 0.71 | 0.42 | 0.64 | 0.55 |
| Pretrained | 6 | 0.63 | 0.46 | 0.44 | 0.52 |

**Prediction Results and Interpretation**

- From visual inspection the Scratch model seems to predict mostly areas in yellow of high extent and high boundary probabilities, while the Pretrained model predicts areas of high extent and high distance probabilities in magenta (Figure 34).
- Visual inspection suggests that Scratch predictions more strongly identifies both field interiors and field edges, while the Pretrained model locates field extents but is less confident about the exact locations of boundaries (Figure 34).
- From the subset of PA1, some square shapes can be seen which can be fields. This may mean that the predictions are detecting some sort of field structures (Figure 35).
- No clear field boundaries can be seen for the Scratch or Pretrained predictions as could be for the Wallonia predictions (Figure 35).

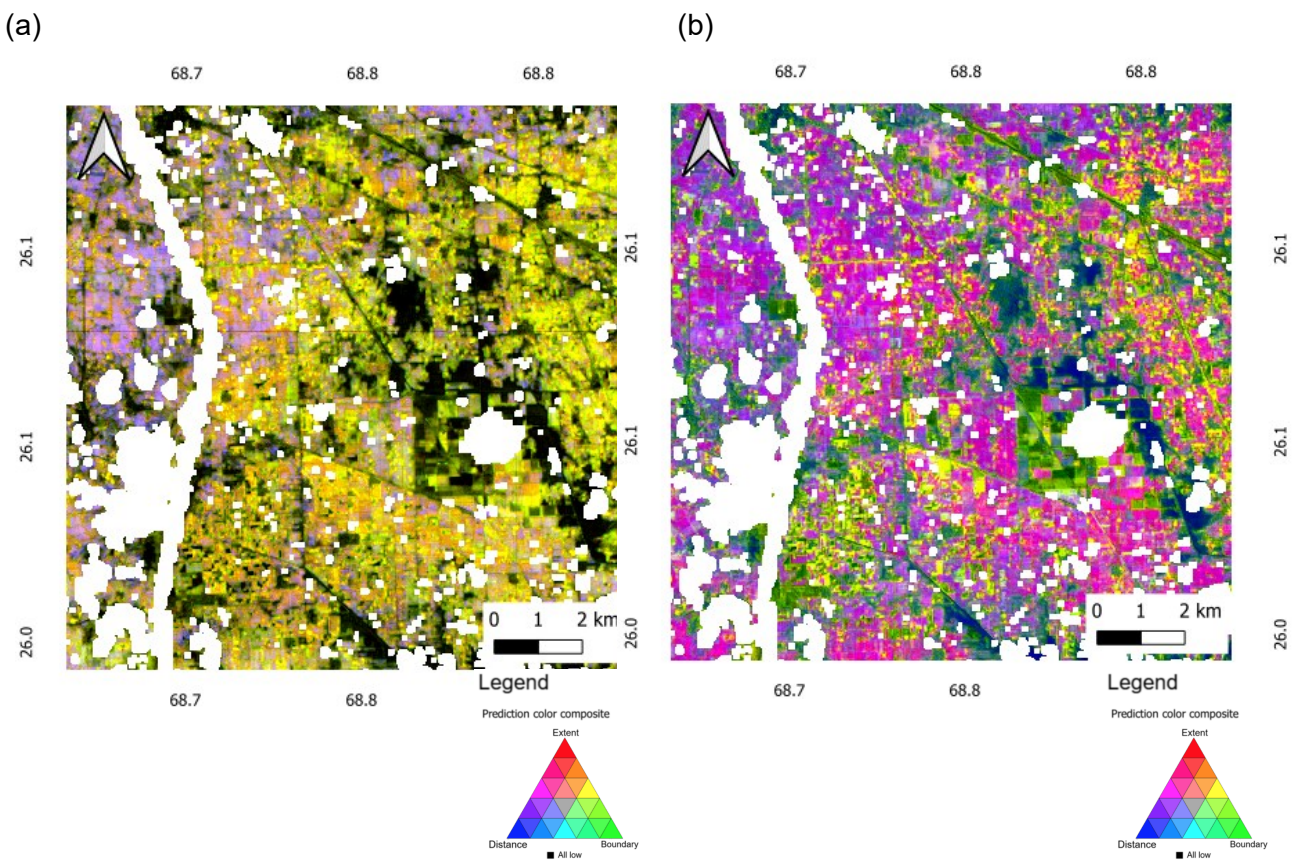



Figure 36: Scratch (a) and Pretrained (b) model predictions in PA1.

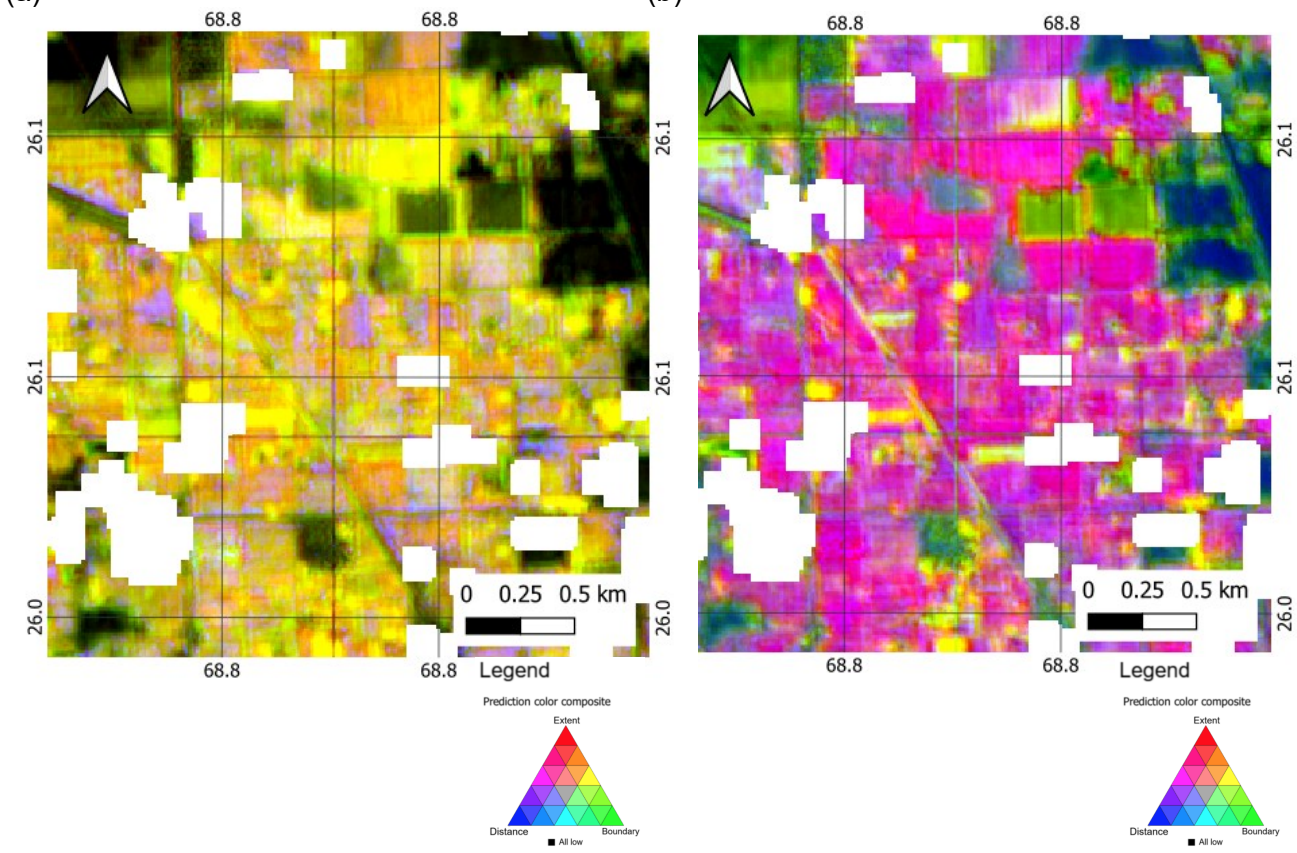


Figure 36: Scratch (a)  and Pretrained (b) model predictions in subset of PA1.

**Vectorization Results and Interpretation**

- It is not clear if the vectorization code is correctly segmenting what it thinks to be fields based on the parameters it is given, or if those parameters need to be adjusted (Figure 36).
- There is no indication that the vectorized predictions for the Scratch and Pretrained models fall in line with the in-situ segmentation (Figure 37).

(a) (b)

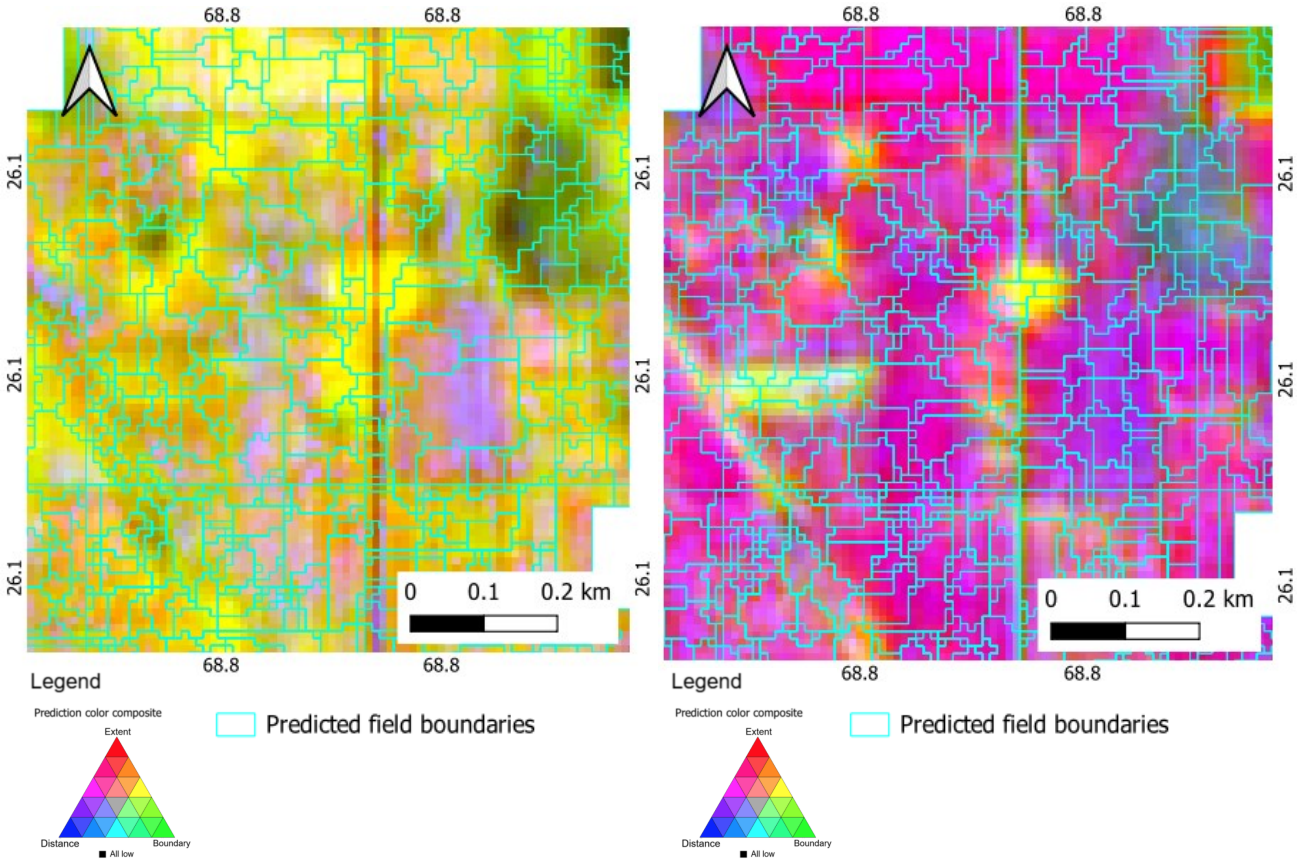



Figure 38: Scratch (a) and Pretrained (b) prediction and vectorized field boundaries.

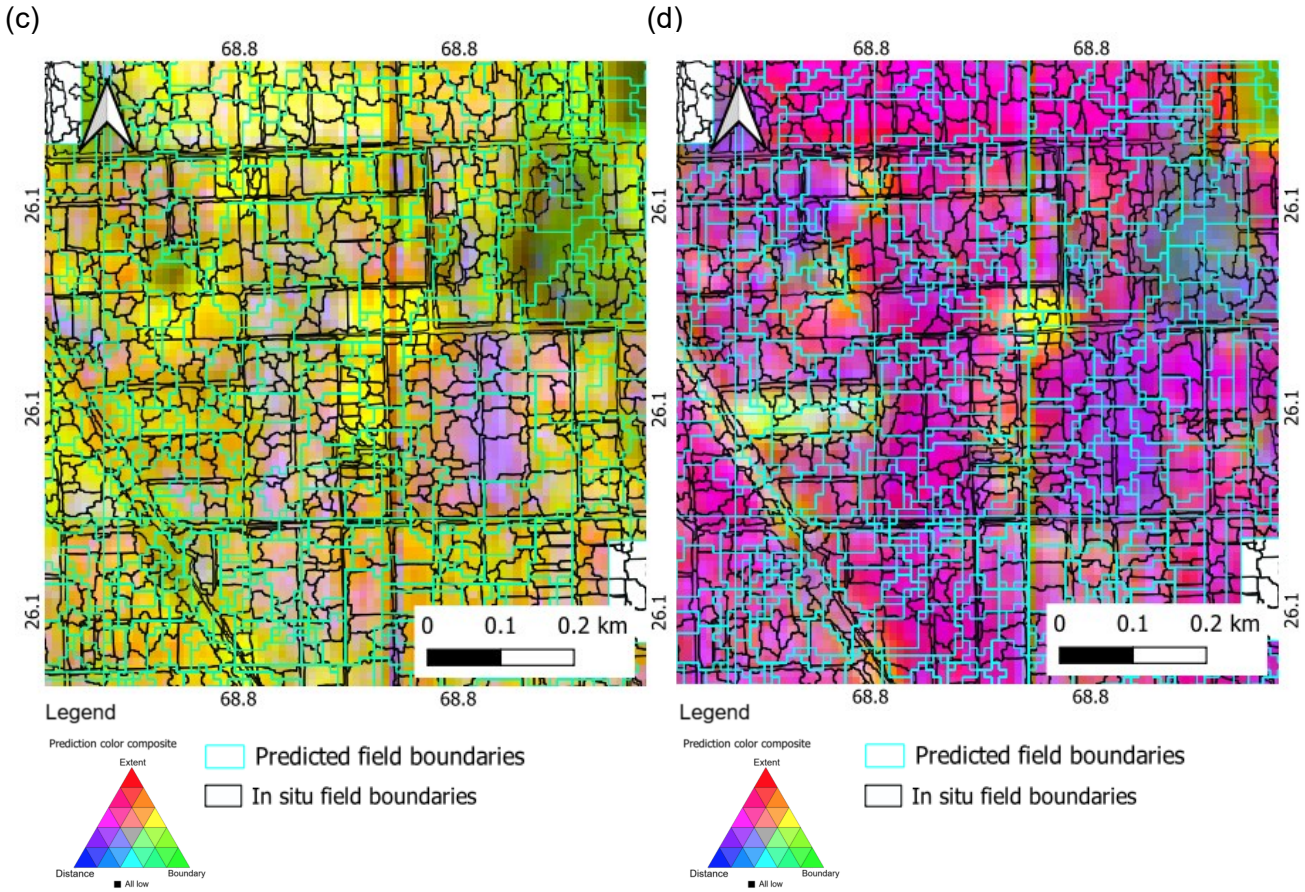



Figure 39: Scratch (a) and Pretrained (b) prediction and vectorized field boundaries with in-situ field boundaries.

**Evaluation Metrics Results and Interpretation:**

Table 12: Evaluation Metrics of Predictions in Pakistan area

| Model | MCC | Median IoU | IoU ≥ 0.5 |
|---|---|---|---|
| Scratch | 0.092938 | 0.012033 | 0.000002 |
| Pretrained | 0.092976 | 0.010271 | 0.000024 |

- Very low MCC indicates only a weak correlation between segmentations and ground truth.
- It cannot be said if pre-training provides a meaningful improvement.
- Very low IoU demonstrates that the majority of predictions are completely misaligned with ground truth.

**Overall remarks:**

- Pretraining decreases the number of epochs by 88% with relatively similar or slightly worse training and validation loss for this particular study.
- The pretreatment methodology of the Wallonia research cannot be directly transferred to the Pakistan area because in Pakistan:
    - Fields are much smaller
    - There do not seem to be clear boundaries between fields
    - Many crop fields of the same crop are sown together (difficult segmentation)
    - Different growing season and crops
    - Less training and pre-training data are used
    - There are only crop and non-crop areas, no unknown areas like in Wallonia
    - In situ data is a segmentation (not true ground truth data) which carries its own errors and assumptions
- The Wallonia AOI methodology must be adapted to the Pakistan landscape before comparisons between the two areas can be made.
- Seeing as pre-training has an effect in the Pakistan Case Study, the next step can be to pretrain a model in Wallonia and fine tune in Pakistan as performed by Wang, Waldner & Lobell (2022) who pre-trained a CNN on France field boundaries and used labels in India to fine-tune the model to predict in India.
- Visual inspection of the predictions does not clearly indicate whether underlying field boundary patterns have been learned by either model.
- The vectorized predictions are not trustworthy and their evaluation is unsuitable to say anything about model performance.
- In conclusion, pre-training decreases training time and has an effect on model training and prediction, but nothing can be concluded about improved model performance from this case study.

# Chapter 7

# Discussion

## 7.1 Pretraining effect on training

Pretraining reduces the number of epochs needed for the training and validation loss to stabilize for all in-situ datasets but one (4% dataset). This likely occurs as the pre-training already provides the model a preview of the data it will see in the training. While fewer epochs are needed for training, the total time of running the pre-training and subsequently Pretrained models is greater, as pre-training takes considerably more time than training. This is because the extent of pre-training is considerably larger, and many more data layers are used (22 UL segmentations). Therefore, from a time perspective, for this application, pre-training is beneficial if time is not an issue or if the saved Pretrained model is used multiple times.

Additionally, pre-training does reduce the training and validation loss throughout training. Using less in-situ data results in higher training and validation loss, while requiring fewer training epochs. This is to be expected, as there is less data available for the model to learn meaningful patterns from. Additionally, pre-training during more epochs could even further decrease the number of training epochs, but may increase the risk of overfitting

## 7.2 Model Prediction

In terms of the model predictions, all models greatly overestimate the agricultural area generally by more than double the actual value in the LPIS dataset. The Pretrained models exhibit less of an overestimation which is possibly due to having seen the Crop/Non-Crop Classification during pre-training. Given that on the Crop/Non-Crop map, an Overall Accuracy of 0.92 was achieved when comparing the prediction to the crop and non-crop areas in the LPIS for Wallonia, it is possible that a more accurate Crop/Non-Crop Classification could increase the performance of the Pretrained models.

Moreover, the predicted number of agricultural fields for the models is relatively close to the LPIS number of fields, especially for the Scratch models. However, Pretrained models underestimate the number of fields by an average of 19.2%. Additionally, from the 2% to 1% Scratch models, estimated agricultural area greatly increases but number of fields greatly decreases, indicating prediction of less but bigger fields.

Interestingly, after closer investigation of the LPIS dataset, many instances of doubled polygon boundaries are found. For some fields, there is a primary boundary line going around the field and a second boundary line going around part or all of the field boundary at a small distance away. An example of this is shown in Appendix B. For the model's vectorized predictions, only one line around each field is observed. This occurs for all models regardless of dataset percentage, likely because the *duplicated* LPIS boundaries occur at distances below the 10 m spatial resolution of the NDVI composites, making them undetectable within the input data used by the models. It is unlikely that the secondary lines going around fields are fields themselves. It is possible that they are agri-environmental stripes along fields or access

trenches between fields which were not discarded from the LPIS dataset during pre-treatment. In the future, it should be considered whether to discard these fields.

Additionally, both the Pretrained and Scratch models tend to more wrongly predict fields on pastures and meadows than on other land types. This is seen as only 25% of the area of Wallonia is made up of pastures and meadows, but an average of 74% incorrectly predicted field area is on pasture and meadow land across all models. It is possible that this occurs because there are structures that resemble fields like pastures and meadows that are incorrectly delineated as fields. It is also possible that adjacent meadows and pastures are reported as a single block by farmers but are mowed and grazed in smaller parcels within the block, thereby introducing variability and affecting classification outcome. This helps explain the substantial overprediction of fields in non agricultural field areas in the East region of Wallonia. Given that this particular region is different from the rest of Wallonia because there are comparatively a reduced number of fields and an increased number of pastures and meadows, it is suggested to include part of this region as a new training area. This may allow for the model to learn and distinguish the relationship in that zone, but it may have the drawback of confusing the model and generating a worse overall prediction.

This oversegmentation in the West of Wallonia, can help to explain the disproportionately high prediction of agricultural area. However, this paired with the similar and lower number of fields predicted for the Scratch and Pretrained models signals an undersegmentation of field boundaries. After visual inspection, a general undersegmentation is observed to be spread out throughout Wallonia, with the exception of the oversegmentation in the West. It is expected to have some general undersegmentation, as adjacent fields of the same crop are difficult to distinguish at 10 meter spatial resolution without a clear separation like a road or field margin between them. As in-situ data training percentages decrease, it becomes more common for adjacent field boundaries to merge. The lower percentage Scratch predictions contain substantial oversegmentation and contain many very small polygons of incorrectly segmented non-field artifacts. For the Pretrained model predictions, these small polygons are not as common.

Regarding visual interpretation, the Scratch and Pretrained models trained with the 100% dataset produced similar predictions, particularly in high-confidence agricultural regions with clearly defined field boundaries. Differences between the models are mainly observed in lower-confidence and non-agricultural areas. Both models occasionally vectorize uncertain non-agricultural regions into field polygons that are not present in the LPIS dataset, which indicates false-positive field detection. Clear differences are observed between the 100% and 1% Scratch model predictions, showing greater variation within field areas, and a more seemingly uncertain prediction.

In contrast, the 1% Pretrained model greatly resembles the 100% Pretrained model indicating that pre-training allows the CNN to maintain stable segmentation performance under limited in-situ training data conditions. Furthermore, the 1% Scratch model contains bigger field polygons because of merged boundaries, particularly in the West area, which can explain the high peak in field area and extreme drop in number of fields. This indicates that at this percentage, the 1% Scratch model is failing to segment fields properly. For the 1% dataset models, it seems like the general field area and boundaries are captured, however the models struggle with incorrectly segmenting artifacts in non-field areas as fields.

## 7.3 Model Performance

The Scratch model achieves MCC values which range from 0.63 for the 100% dataset, and gradually decrease to 0.41 for the 1% dataset. While the Pretrained model achieves peak performance with the 25% dataset with an MCC of 0.69, which gradually decreases to a minimum of 0.55 on the 2% dataset. These ranges of values indicate moderate to strong agreement between the predicted and actual values for both models, with slightly stronger agreement for the Pretrained models. This illustrates that at a pixel level, the classification between field boundaries and non field boundaries is generally effective. Notably, a steep decrease in MCC is experienced for the Scratch model when going from the 2% dataset to the 1% dataset.

The Pretrained and Scratch models do not exhibit differences between each other regarding the median IoU and IoU $\geq$ 0.5. The median IoU ranges from 0.24 to 0.11 for the Scratch models, and 0.26 to 0.18 for the Pretrained. While, for the IoU $\geq$ 0.5, the Scratch models score from 0.31 to 0.16 and for the Pretrained models from 0.32 to 0.23. Unlike the Scratch model, the Pretrained model does not experience an abrupt collapse in IoU performance at 1%, suggesting that pre-training helps preserve object-level segmentation capability when only minimal in-situ data are available

The relatively low median IoU, combined with only 32% to 16% of polygons exceeding the 0.5 threshold, indicates that while a subset of fields are delineated with acceptable accuracy which is greater than or equal to the set threshold, the majority of predictions exhibit poor geometric agreement with reference boundaries. Additionally, after looking at the MCC and IoU metrics, it seems that pre-training improves the CNN's capacity to discriminate between field and non-field regions, particularly under limited in-situ data conditions. However pre-training does not substantially enhance the geometric accuracy of field boundary delineation.

The evaluation of the vectorized predictions restricted to agricultural areas primarily serves to exclude the impact of false boundary predictions in non-agricultural regions. However, by clipping the vectorized predictions to the LPIS agricultural field extents, the outer boundaries of the clipped areas cut off predicted polygons extending the agricultural extent and actually become the borders of the predicted polygons. As a result, the number of true positive boundary pixels is artificially increased. This, along with eliminating many false positive predictions of field boundaries in non-agricultural areas, render the MCC metric essentially unusable to evaluate model predictions in this case. This is also why the MCC exhibits near perfect alignment of 0.98 to 0.99 across all model runs. Additionally, the general increase in IoU metrics for both model types from the full Wallonia extent to the agricultural fields extent indicates an improved geometric agreement between predicted and reference field polygons of performing this operation.

Thereby, in a future application, it is suggested to evaluate the model predictions constrained to agricultural areas, as evaluation metrics may increase. However, an improvement is to perform the clip with a land cover map of rougher resolution, as to not artificially create true boundaries that the model did not predict. Additionally, a land cover map is likely to exist in regions with limited in situ data and can be used to discriminate against some land cover classes to improve prediction post-processing.

A related study by Wang, Waldner, and Lobell (2022) investigated crop field boundary delineation using a CNN model pretrained on agricultural parcels in France and fine-tuned in India. A median IoU of 0.52 was achieved when using PlanetScope imagery with a spatial resolution of 1.5m and a median IoU of 0.85 when using Airbus SPOT imagery at a spatial resolution of 4.8m. The corresponding MCC values were 0.52 and 0.51 respectively. The higher IoU values obtained in their study may be due to the finer spatial resolution of the input imagery compared to the 10 m resolution used in the present study. Additional factors, including differences in methodology, cloud presence, image quality, and model inputs, may also have contributed to the performance gap. A further difference is that the study did not employ EL for pre-training, instead, approximately 10,000 field boundaries were manually delineated in India for fine-tuning and evaluation (Wang, Waldner & Lobell, 2022).

An additional aspect is that the MCC value for the Pretrained model increases from 0.55 for the 2% dataset to 0.65 for the 1% dataset. This increase is unexpected, as it would be expected for model performance to further decrease with reduced amounts of in-situ training data. However, there are some random factors which may influence model performance at any given model run, and could account for this variation in performance. For example, it is possible that the random selection of in-situ data samples and the sequence of patches used during pre-training and training could favour or impair the training process. This could be seen reflected in the predictions, with a particularly strong or weak performing segmentation. These factors could have favoured the 1% and discriminated against the 2% dataset.

In order to obtain a better understanding of these random factors' effect on model performance, they could be quantified. To do this, multiple subsets of in-situ data containing a different random selection of field polygons could be made for each dataset percentage considered. Running the model on multiple independently sampled datasets of each percentage would enable calculating the mean and standard deviation of the multiple runs, which could give an understanding and reduction of the randomness. The same could be done in a standardized way with the order of the training patches to diminish its effect. Due to a time constraint, this analysis could not be performed within the scope of this study but could improve result quality.

It is worth mentioning that the vectorization procedure can have a large influence on evaluation results. Since thresholds are used to turn the three band prediction into a vector, certain parameter configurations can be better suited to segment correctly certain in-situ data percentages. Additionally, some parameters may produce results that benefit or penalize certain evaluation metrics by influencing polygon shape, boundary precision, and the degree of fragmentation or merging in the predicted field geometries.

It is possible that the relatively low IoU values are driven either by limitations in the model's ability to accurately delineate field boundaries or by the vectorization code to turn those predictions into polygon boundaries. As a possible workaround, the vectorized predictions could be buffered, to essentially increase polygon width. This could increase the probability of overlap with the actual LPIS field boundaries. This could produce a map with more overlap of predicted and actual field boundaries but with a coarser spatial precision. Nevertheless, depending on the subsequent application it can be a valid map, particularly in in-situ data poor regions.

A significant factor which affected feature creation and likely hindered model performance, is that the study year 2024 was an unusually cloudy year in the study area. This affected the quantity and quality of NDVI composites which served as model input features. Under ideal conditions with lower cloud cover it would have been possible to generate a greater number of monthly NDVI composites, using fewer acquisition dates per composite. Thus improving temporal consistency and image quality. Cloud gaps in the composites can impair correct segmentation in and around gaps by creating incomplete or fragmented field boundaries. Although image acquisitions and training patches with excessive cloud cover were excluded, relatively lenient filtering criteria had to be applied, as stricter thresholds would have resulted in an insufficient amount of training data. While this is unfortunate, the primary objective of this study was not to achieve the highest possible segmentation accuracy, but rather to compare the performance of Scratch-trained and Pretrained model methodologies under varying amounts of training data.

Additionally, due to high cloud cover, particularly in TA1, more than half of training patches were discarded in that area. However, the selection of training polygons was done on the three entire training areas. This means that each percentage dataset containing less than 100% in-situ data, was actually operating with a slightly lower percentage of in-situ data. In the future, the order of steps should be inverted, and polygon selection should be performed after valid patches are identified, in order to have accurate in-situ data percentages.

# Chapter 8

# Conclusion

The objective of this thesis was to determine the effect of pre-training a CNN on the amount of in-situ data needed to perform agricultural field delineation. For this, two model methodologies were developed, the Scratch and Pretrained models. Each model methodology was implemented on datasets of decreasing size of in-situ data for the Scratch model, which received no pre-training and the Pretrained model.

An effect of pre-training is that it decreases the number of epochs required for the training and validation loss to stabilize compared to the Scratch models, meaning that model training was done faster. An average reduction in epochs of 23% from the Pretrained to Scratch models was experienced. Additionally, the Pretrained model generally reduced the training and validation loss at the beginning, during and at the end of training.

Both models vastly overestimate field area compared to the LPIS in-situ dataset, with it more than doubling for most runs. The Pretrained model experiences a slightly less drastic overestimation. In terms of the predicted number of fields the Scratch models are generally closer to the LPIS value, but the Pretrained models exhibit an underestimation averaging 19.2% less agriculture fields. A likely explanation of these over and under estimations, is the oversegmentation of fields in the West area of Wallonia which is present for all model runs.

It would be interesting to determine where and why these over and under segmentations happen to understand why the model is failing in this regard. Particularly since a predicted larger agriculture area with a reduced number of fields signifies less but bigger field polygons than are present in the LPIS, but this does not seem to be the case in the predictions. Discovering this is relevant for future work, seeing as, should the root cause of this be less accurate segmentations in certain areas, those areas can be included in the training of the model. On the other hand, if the issue is that the model consistently overestimates field boundary extents, then the CNN configuration can be altered to correct this.

The evaluation results indicate that the Pretrained models consistently outperform the models trained from Scratch. This improvement is reflected in the classification performance, suggesting that pre-training enables a more accurate distinction between field and non-field pixels. Despite these gains, the delineation of field boundaries remains challenging, as object-based metrics reveal limitations in accurately capturing field extents and boundary geometry. Although the Pretrained models achieve superior performance overall, the evaluation metrics suggest that further improvements are necessary before the models can be considered sufficiently reliable for operational applications.

Based on the predicted area and number of fields, the visual interpretation and the model evaluation metrics, model performance for the Scratch models starts to rapidly decrease when using 2% in-situ training data. For the Pretrained model, model failure is not evident around

2%, and while evaluation metrics steadily decrease with decreasing in-situ data use, there is not a steep decline in performance.

Now that the effectiveness of the pre-training methodology has been validated, future research can focus on further improving model performance. Particularly, there can be value to investigating the effect that the different UL segmentations used have on model performance. Such analyses could be useful in order to keep combinations of UL segmentations that offer information and improve model performance while avoiding overfitting. Additionally, the effect of different EL methods on the UL segmentations can also be investigated.

Furthermore, pre-training does reduce the issue of oversegmentation of fields in non field areas possibly because of having been exposed to the Crop/Non-Crop map in the pre-training. Therefore, pre-training can be a strategy to mitigate oversegmentation. As such, generating and using a Crop/Non-Crop map with a greater Overall Accuracy (currently 0.92) could help improve performance in future re-runs. Future studies can investigate the generation of multiple Crop/Non-Crop prediction maps and evaluate the impact of applying ensemble learning techniques to combine these predictions on overall model performance.

Additionally, the effect of pre-training for longer than one epoch is another viable route of future research. This can allow for the model to find more suitable weights and therefore reinforcing the advantages of pre-training, while facing the risk of overfitting.

Moreover, by performing the Case Study on Pakistan, it was shown that pre-training using EL decreases the number of training epochs used and that pre-training has an effect on the data. While, it is not possible to say more on pre-training effectiveness, the Case Study positions Pakistan as a candidate for applying the findings of the Wallonia research on a region with limited in-situ data. It is evident that the methodology needs to be adapted to consider differences in the Pakistan agricultural system, but future research can continue on aspects like transferring the learned weights from Wallonia to Pakistan models or performing pre-training in Wallonia and fine-tuning the model in Pakistan.

Overall, pre-training using ensemble-based unsupervised learning improves CNN performance for agricultural field boundary detection, particularly under conditions of limited in-situ data. The Pretrained models maintained performance under reduced training data conditions where Scratch models exhibited clear performance degradation, demonstrating the potential of pre-training to reduce dependence on extensive in-situ datasets. Future work should focus on improving boundary delineation accuracy, investigating the influence of different unsupervised learning segmentations and ensemble learning strategies, refining Crop/Non-Crop pre-training datasets, and exploring extended pre-training schemes to further enhance model robustness and generalization performance.

# Sources


Agrawal, H., & Desai, K. (2024). *Canny edge detection: A comprehensive review*. *International Journal of Technical Research & Science, 9*(Spl), 27–35. https://doi.org/10.30780/specialissue-ISET-2024/023

Anwar, H., Qamar, U., & Qureshi, A. W. M. (2014). *Global optimization ensemble model for classification methods*. *The Scientific World Journal, 2014*, Article 313164. https://doi.org/10.1155/2014/313164

Asam, S., Eisfelder, C., Hirner, A., Reiners, P., Holzwarth, S., & Bachmann, M. (2023). AVHRR NDVI compositing method comparison and generation of multi-decadal time series—A TIMELINE thematic processor. *Remote Sensing, 15*(6), Article 1631. https://doi.org/10.3390/rs15061631

Ataei, M., Bussmann, M., Shaayegan, V., Costa, F., Han, S., & Park, C. B. (2020). *NPLIC: A machine learning approach to piecewise linear interface construction*. arXiv. https://arxiv.org/abs/2007.04244

Ayeni, J. A. (2022). *Convolutional neural network (CNN): The architecture and applications.* Applied Journal of Physical Science, 4(4), 42-50. https://doi.org/10.31248/AJPS2022.085

Bassine, C., Radoux, J., Beaumont, B., Grippa, T., Lennert, M., Champagne, C., De Vroey, M., Martinet, A., Bouchez, O., Deffense, N., Hallot, E., Wolff, E., & Defourny, P. (2020). *First 1-m resolution land cover map labeling the overlap in the 3rd dimension: The 2018 map for Wallonia*. *Data, 5*(4), 117. https://doi.org/10.3390/data5040117

Bezdan, T., & Džakula, N. B. (2019). Convolutional neural network layers and architectures. In *Proceedings of the International Scientific Conference on Information Technology and Data Related Research (SINTEZA 2019)* (pp. 445-451). https://doi.org/10.15308/Sinteza-2019-445-451

Chicco, D., & Jurman, G. (2020). *The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation*. *BMC Genomics, 21*(1). https://doi.org/10.1186/s12864-019-6413-7

Claverie, M., Ju, J., Masek, J. G., Dungan, J. L., Vermote, E. F., Roger, J.-C., Skakun, S. V., & Justice, C. (2018). *The harmonized Landsat and Sentinel-2 surface reflectance data set*. *Remote Sensing of Environment, 219*, 145–161. https://doi.org/10.1016/j.rse.2018.09.002

Ding, K., Xue, L., Ran, X., Wang, J., & Yan, Q. (2023). CNN2D-SENet-based prospecting prediction method: A case study from the Cu deposits in the Zhunuo mineral concentrate area in Tibet. *Minerals, 13*(6), 730. https://doi.org/10.3390/min13060730

European Commission. (n.d.). *Managing payments*. European Commission – Common Agricultural Policy. https://agriculture.ec.europa.eu/common-agricultural-policy/financing-cap/assurance-and-audit/managing-payments_en

European Space Agency. (2015). *Sentinel-2 user handbook*. European Space Agency. https://sentinels.copernicus.eu/documents/247904/685211/Sentinel-2_User_Handbook

Gobin, A., Sallah, A.-H. M., Curnel, Y., Delvoye, C., Weiss, M., Wellens, J., Piccard, I., Planchon, V., Tychon, B., Goffart, J.-P., & Defourny, P. (2023). *Crop phenology modelling using proximal and satellite sensor data*. *Remote Sensing, 15*(8), 2090. https://doi.org/10.3390/rs15082090

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., Han, W., Huang, M., Jin, Q., Lan, Y., Liu, Y., Liu, Z., Lu, Z., Qiu, X., Song, R., & Tang, J. (2021). *Pre-trained models: Past, present and future*. *AI Open, 2*, 225–250. https://doi.org/10.1016/j.aiopen.2021.08.002

He, K., Girshick, R., & Dollár, P. (2019). *Rethinking ImageNet pre-training*. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019) (pp. 4918–4927). IEEE. https://doi.org/10.1109/ICCV.2019.00502

Hussein, B. M., & Shareef, M. M. (2024). *An empirical study on the correlation between early stopping patience and epochs in deep learning*. *ITM Web of Conferences, 64*, Article 01003. https://doi.org/10.1051/itmconf/20246401003

Islam, M., Chen, G., & Jin, S. (2019). *An overview of neural network*. *American Journal of Neural Networks and Applications, 5*(1), 7–11. https://doi.org/10.11648/j.ajnna.20190501.12

Jadama, A. F., Jobarteh, B., Islam, M. M., & Toray, M. K. (2024). *Ensemble learning: Methods, techniques, application*. ResearchGate. https://doi.org/10.13140/RG.2.2.28017.08802

Korchagina, I. A., Goleva, G., Savchenko, Y., Savchenko, Y., & Bozhikov, T. S. (2020). *The use of geographic information systems for forest monitoring*. Journal of Physics: Conference Series, 1515(3), 032077. https://doi.org/10.1088/1742-6596/1515/3/032077

LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Li, M. (2024). *Comprehensive review of backpropagation neural networks*. *Academic Journal of Science and Technology, 9*(1), 150–154. https://doi.org/10.54097/51y16r47

Lombardi, A., Ascari, F., & Brusa, E. (2021). *Stress-strain evaluation of structural parts using artificial neural networks. Engineering Applications of Artificial Intelligence, 103*, 104300. https://doi.org/10.1016/j.engappai.2021.104300

Mei, W., Wang, H., Fouhey, D., Zhou, W., Hinks, I., Gray, J. M., van Berkel, D., & Jain, M. (2022). Using deep learning and very-high-resolution imagery to map smallholder field boundaries. *Remote Sensing, 14*(13), 3046. https://doi.org/10.3390/rs14133046

Mienye, I. D., & Swart, T. G. (2024). *A comprehensive review of deep learning: Architectures, recent advances, and applications*. *Information, 15*(12), 755. https://doi.org/10.3390/info15120755

Miseta, T., Fodor, A., & Vathy-Fogarassy, Á. (2024). *Surpassing early stopping: A novel correlation-based stopping criterion for neural networks*. *Neurocomputing, 567*, 127028. https://doi.org/10.1016/j.neucom.2024.127028

Mohammed, A., & Kora, R. (2023). *A comprehensive review on ensemble deep learning: Opportunities and challenges*. *Journal of King Saud University – Computer and Information Sciences, 35*(2), 757–774. https://doi.org/10.1016/j.jksuci.2023.01.014

Molina, J. M., Llerena, J. P., Usero, L., & Patricio, M. A. (2025). *Advances in instance segmentation: Technologies, metrics and applications in computer vision*. *Neurocomputing, 625*, 129584. https://doi.org/10.1016/j.neucom.2025.129584

Montesinos-López, O. A., Montesinos-López, A., & Crossa, J. (2022). *Fundamentals of Artificial Neural Networks and Deep Learning.* In *Multivariate Statistical Machine Learning Methods for Genomic Prediction* (pp. 379–425). Springer. https://doi.org/10.1007/978-3-030-89010-0_10

Nouna, A., Nouna, S., Mansouri, M., & Boujemâa, A. (2025). *Deep 2D convolutional neural network architecture for hyperspectral land cover classification: A comparative study with KNN*. Informatica, 49(34). https://doi.org/10.31449/inf.v49i34.9480

Padilla, Rafael, Sergio L. Netto, and Eduardo A. B. da Silva (2020). “A Survey on Per formance Metrics for Object-Detection Algorithms”. In: pp. 237–242. DOI: 10.1109/IWSSIP48289.2020.9145130.

Pires, P. B., Santos, J. D., & Pereira, I. V. (2023). *Artificial neural networks: History and state of the art*. In *Encyclopedia of Information Science and Technology* (6th ed.). IGI Global. https://doi.org/10.4018/978-1-6684-7366-5.ch037

Prabu, S., & Gnanasekar, J. M. (2021). *A study on image segmentation method for image processing*. *Advances in Parallel Computing, 38*, 223–229. https://doi.org/10.3233/APC210223

Pu, J., Zhao, S., Cheng, L., Chang, Y., Wu, R., Lv, T., & Zhang, R. (2023). *Examining the effect of pre-training on time series classification*. arXiv. https://doi.org/10.48550/arXiv.2309.05256

scikit-image developers. (n.d.). *skimage.filters.laplace*. In *scikit-image documentation (stable version)*. https://scikit-image.org/docs/stable/api/skimage.filters.html#skimage.filters.laplace

Sekhar, C., & Sai Meghana, P. (2020). *A study on backpropagation in artificial neural networks. Asia-Pacific Journal of Neural Networks and Its Applications, 4*(1), 21–28. https://doi.org/10.21742/AJNNIA.2020.4.1.03

Service public de Wallonie. (n.d.). *Géoportail de la Wallonie*. https://geoportail.wallonie.be/walous

Bozinovski, S. (2020). *Reminder of the first paper on transfer learning in neural networks, 1976. Informatica, 44*(3) 291–302 . https://doi.org/10.31449/inf.v44i3.2828

Sun, Z., Wang, G., Li, P., Wang, H., Zhang, M., & Liang, X. (2024). *An improved random forest based on the classification accuracy and correlation measurement of decision trees. Expert Systems with Applications, 237* (Part B), 121549. https://doi.org/10.1016/j.eswa.2023.121549

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). *A survey on deep transfer learning*. In Artificial Neural Networks and Machine Learning – ICANN 2018 (Lecture Notes in Computer Science, Vol. 11141, pp. 270–279). Springer. https://doi.org/10.1007/978-3-030-01424-7_27

Wang, S., Waldner, F., & Lobell, D. B. (2022). *Unlocking large-scale crop field delineation in smallholder farming systems with transfer learning and weak supervision. Remote Sensing, 14*(22), 5738. https://doi.org/10.3390/rs14225738

Wen, T., Tong, B., Liu, Y., Pan, T., Du, Y., Chen, Y., & Zhang, S. (2022). *Review of research on the instance segmentation of cell images*. Computer Methods and Programs in Biomedicine, 227, 107211. https://doi.org/10.1016/j.cmpb.2022.107211

Wołk, K., & Tatara, M. S. (2024). *A review of semantic segmentation and instance segmentation techniques in forestry using LiDAR and imagery data. Electronics*, 13(20), 4139. https://doi.org/10.3390/electronics13204139

Wurm, M., Stark, T., Zhu, X. X., Weigand, M., & Taubenböck, H. (2019). *Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. ISPRS Journal of Photogrammetry and Remote Sensing, 150*, 59–69. https://doi.org/10.1016/j.isprsjprs.2019.02.006

Xu, H., & Xiao, Y. (2018). *A novel edge detection method based on the regularized Laplacian operation. Symmetry, 10*(12), 697. https://doi.org/10.3390/sym10120697

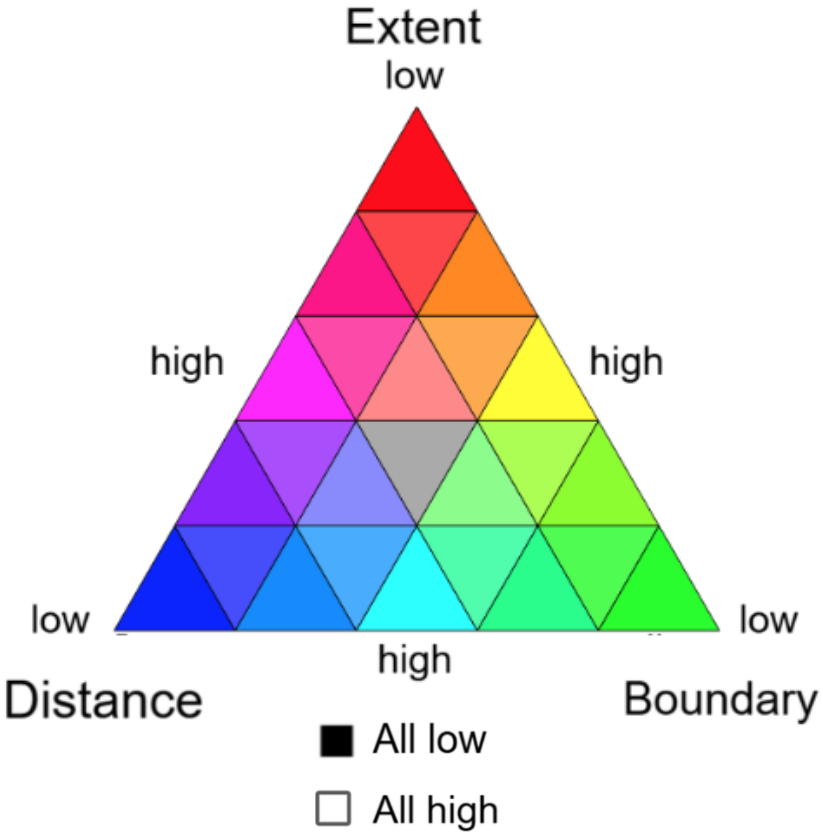
Yang, Y., Lv, H., & Chen, N. (2023). *A survey on ensemble learning under the era of deep learning. Artificial Intelligence Review, 56*(6), 5545–5589. https://doi.org/10.1007/s10462-022-10283-5

Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., & Gao, M. (2023). *Techniques and challenges of image segmentation: A review. Electronics, 12*(5), Article 1199. https://doi.org/10.3390/electronics12051199

Zhang, Y., Ma, J., Liang, S., Li, X., & Liu, J. (2022a). *A stacking ensemble algorithm for improving the biases of forest aboveground biomass estimations from multiple remotely sensed datasets. GIScience & Remote Sensing, 59*(1), 234-249. https://doi.org/10.1080/15481603.2021.2023842

Zhang, Y., Liu, J., & Shen, W. (2022b). *A review of ensemble learning algorithms used in remote sensing applications. Applied Sciences, 12*(17), 8654. https://doi.org/10.3390/app12178654

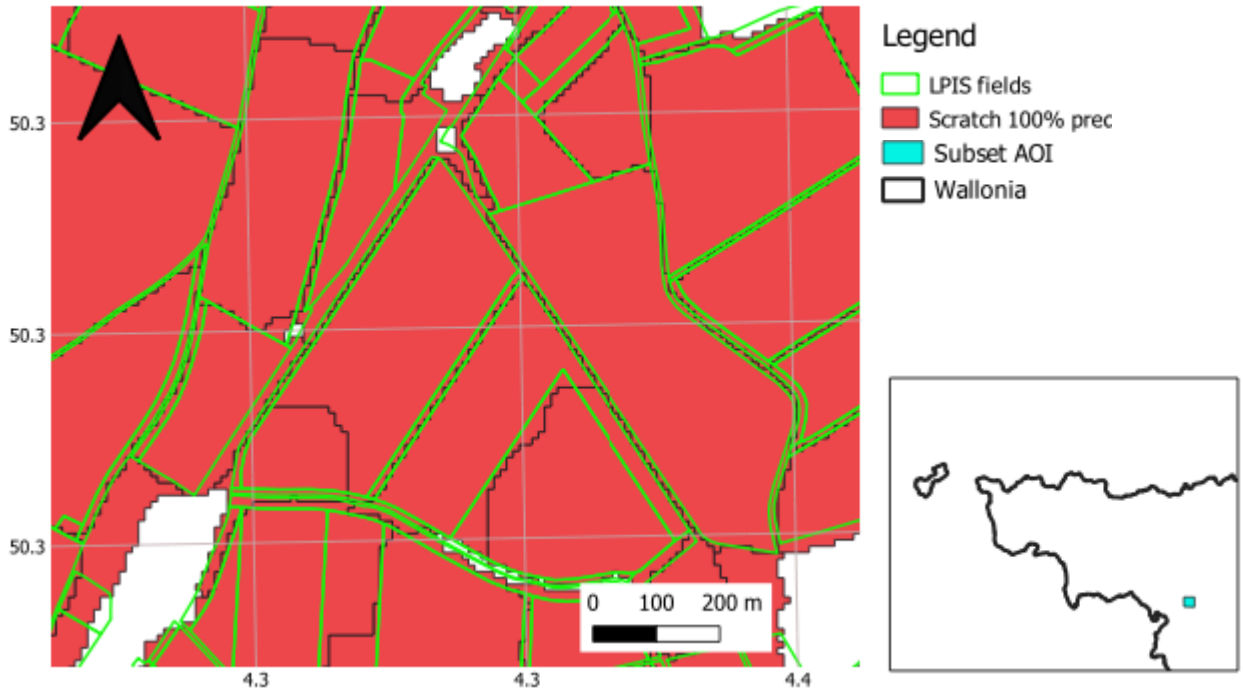
# Appendix A: Ternary RGB Legend for Color Composite Prediction Visualization




# Appendix B: Field Area in Incorrectly Predicted Land Types Across Dataset Percentages

| Field Area | 1% | 2% | 3% | 4% | 5% | 10% | 25% | 50% | 100% |
|---|---|---|---|---|---|---|---|---|---|
| Pastures and Meadows | 84% | 78% | 70% | 72% | 73% | 75% | 70% | 73% | 76% |
| Others | 16% | 22% | 30% | 29% | 27% | 25% | 30% | 27% | 24% |

# Appendix C: Example of Double Boundary Fields in LPIS Dataset



**How does pre-training a Convolutional Neural Network decrease the amount of in-situ data needed to perform agricultural field delineation?**

Natàlia Dinwoodie Palmés

Accurate agricultural field boundary delineation is essential for crop monitoring, yield estimation, and sustainable land management, yet in-situ reference data are often scarce, expensive to obtain or unavailable. CNN models offer a powerful approach for deriving such information from RS data, however, their performance typically depends on large amounts of labelled training data. In data scarce regions, there is a need for strategies to reduce reliance on in-situ data while maintaining reliable segmentation accuracy. Pre-training a model has been shown to decrease training time and improve model performance in some applications, particularly under limited in-situ data conditions. EL approaches have also demonstrated strong potential for improving segmentation quality and can be used to generate unsupervised field boundary information suitable for pre-training. However, the effect of pre-training a CNN using EL for the application of agricultural field boundary delineation remains largely unexplored in literature. This study contributes to addressing this gap.

In this study, two modelling approaches are compared, a Scratch model trained solely on labelled data and a Pretrained model pretrained on EL derived segmentation products. Both are applied under progressively reduced in-situ training datasets. Pre-training reduces the number of training epochs by an average of 23% compared to the Scratch models. Pre-training also improves pixel-based performance, with MCC increasing from 0.63 (best Scratch model, 100% dataset) to 0.69 (Pretrained, 25% dataset) and from 0.41 (1% Scratch) to 0.55 (2% Pretrained). In contrast, improvements in object-based metrics are less pronounced, with IoU ≥ 0.5 and median IoU increasing slightly from 0.16 and 0.11 (Scratch) to 0.23 and 0.18 (Pretrained), respectively. Overall, the benefits of pre-training are most evident under conditions of limited in-situ data.

Overall, EL based pre-training enhances CNN performance to limited amounts of in-situ data and improves field and non-field discrimination, but further work is required to refine boundary precision and reduce oversegmentation. The approach shows potential for application in data-scarce regions where high-quality field boundary datasets are unavailable.

Key words: Ensemble Learning, Model Pre-training, Field Boundary Delineation, Convolutional Neural Network, In-situ data

**UNIVERSITÉ CATHOLIQUE DE LOUVAIN**
**Faculté des bioingénieurs**
Croix du Sud, 2 bte L7.05.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/agro**