HEIDELBERG UNIVERSITY
FACULTY FOR CHEMISTRY AND EARTH SCIENCES
INSTITUTE OF GEOGRAPHY

Fusing OpenStreetMap and Copernicus Sentinel-2 data for large-scale land use and land cover applications using Deep Learning

Master Thesis

Author

Janek Lukas Voß

Supervisors

Prof. Dr. Alexander Zipf
Dr. Sven Lautenbach

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

27 NOVEMBER 2019

M.SC. GEOGRAPHY

MATRICULATION NUMBER: 3479818

E-MAIL: JANEK.VOSS@MAIL.DE

Contents

List of Figures

List of Tables

Acknowledgements

This thesis in its present form would not have been possible if it wasn't for the continuous contribution and advice of Dr. Michael Schultz from the Institute of Geography at Heidelberg University. I am very grateful for his honest feedback, different perspectives and frequent efforts over the course of this thesis. In the last years of my studies I very much enjoyed working with him at the institute where I was able to learn a lot. I would like to acknowledge the advice and support from members of the HeiGIT gGmbH at Heidelberg University, namely Dr. Michael Auer, Rafael Troilo and Fabian Kowatsch. I would also like to thank Dr. Sven Lautenbach from the Institute of Geography for promoting progress and adding valuable inputs to my thesis. Last but not least, I would like to thank my family and friends for their constant support. Special thanks go to Matthias Humt and Simon Busch, who helped me at different stages of the thesis.

Abstract

OpenStreetMap (OSM) data and Sentinel-2 (S2) satellite images were combined to derive land use and land cover (LULC) for a large area in Europe using state-of-the-art Deep Learning (DL) technologies. Training data was synthetized by deriving a classification from OSM features similar to Corine Land Cover (CLC) level 2 and in parts complemented with S2 images from the meteorological summer of 2018. Data preparation, setup and training enables the application of a Fully Convolutional Network (FCN), using the Python Deep Learning library Keras and the Machine Learning (ML) platform Tensorflow. Once trained, the FCN was applied in an automated workflow to produce LULC maps with 10m spatial resolution and temporal and spatial flexibility. Results were subjected to an accuracy assessment and achieve overall accuracies of 62.2% for the study area and 82.9% for a small reference area. However, individual class performances varied largely in terms of map proportions and estimated classification accuracy. The results indicate that large-scale LULC maps created with the proposed approach cannot be considered reliable across the full spectrum of land use, but contain accurate information, depending on certain class memberships. This work identified wide-ranging challenges and offers multiple measures to help improve predictions in the future. Moreover, it illustrates an approach for the fast and simple creation of LULC maps, dealing with cloud cover, seasons and inputs of various sizes. Finally, this thesis proposes a modular, end-to-end workflow and uses open data and open-source software to facilitate reproducibility and continued improvement.

1 Introduction

Over the course of the Holocene (11.650 BC – today) humans have made changes to over 50% of the earth's landmass with accelerating speed. Anthropogenic land use (LU) changes have a decisive impact on the earth's ecosystem, strongly influencing atmospheric conditions, biodiversity, sedimentation and surface characteristics (Waters et al., 2016). Also, land use and land cover (LULC) information play an important role for human societies. They are key components for spatial planning and development, natural resource management, vulnerability and risk management. Environmental applications of LULC information include climate modelling, environmental assessments and monitoring (Jones, 2008). Today, economic as well as political players profit from accurate and up-to-date LULC maps as they help making more informed, evidence-based decisions (Kussul et al., 2017; Thanh Noi and Kappas, 2018).

The surface cover of an area is defined as land cover (e.g. forests, buildings, fields), whereas the purpose of the land is described as its land use (e.g. agriculture, recreation, pasture). Consequently, LULC maps provide an overall picture over human activities and natural elements on the earth's surface (Fisher et al., 2005). There are many existing global, regional and local LULC products with different degrees of topicality, spatial resolution, accuracy and complexity. These include global products, such as GlobeLand30 and GlobCover, as well as regional and continental products, such as Corine Land Cover (CLC), Urban Atlas (UA) and the National Land Cover Database (NLCD).

Producing and validating LULC maps can be costly, since datasets are often created by collecting and interpreting information in the field alongside remote sensing (RS) data (Lavreniuk, 2017; Ndikumana et al., 2018). The classification process normally requires manual or semi-automatic interpretation of remote sensing (RS) images, carried out by experts (Kussul et al., 2017; Nguyen et al., 2018; Thanh Noi and Kappas, 2018). Thus, these products suffer from long update cycles and coarse spatial resolution (Fonte et al., 2016).

To compensate for those shortcomings, Web 2.0 based applications from the area of Volunteered Geographical Information (VGI) were proposed (Sui et al., 2012). Supported by millions of volunteers, the OpenStreetMap (OSM) project provides open source spatial information on a global scale (Wiki, 2019a). In combination with numerous new RS datasets (Kussul et al., 2017) studies confirm high potential of OSM data as a source for LULC maps (Estima and Painho, 2013; Fonte et al., 2017; Schultz et al., 2017).

Nevertheless, using OSM data for LULC applications involves several challenges. In particular, these include overlapping geometries, incorrect object descriptions (tags), temporal inhomogeneity and spatial gaps (Fonte et al., 2016; Schultz et al., 2017). Quality and integrity of OSM data largely depend on the contributor's activity, which can make OSM data unusable for LULC products (Arsanjani et al., 2015). Comparing official LULC products (e.g. CLC) with those derived from OSM can be challenging, since harmonizing both nomenclatures entirely is often very difficult (Arsanjani and Vaz, 2015; Estima and Painho, 2013; Schultz et al., 2017).

The goal of this thesis is to design and examine a Deep Learning (DL) model in order to predict LULC classes, using Sentinel-2 (S2) imagery in combination with OSM data. Also, issues regarding existing LULC products are addressed. First, the proposed approach allows for an automatic generation of LULC maps with temporal

flexibility and spatial transferability. It resolves spatial incompleteness of raw OSM data by integrating S2 images to create a more detailed product. Lastly, this product is comparable with CLC and UA, using large parts of their nomenclatures.

1.1 Research Questions

In this thesis, the following research questions are addressed:

1. *How can 10m RGB Sentinel 2 data and OSM features be combined within a Deep Learning framework to obtain a land use classification?*

The workflow presented in the following chapter (Chapter 2) illustrates one way of implementing such a task. Every step leading to this specific workflow incorporates considering and justifying different choices.

2. *What is the suitability of OSM features for LULC mapping?*

To address this question, an accuracy assessment of the resulting classification is performed. Using a confusion matrix, overall accuracy, producer's accuracies and user accuracies are estimated.

2 Methods

In this chapter data sources, classification methods and key concepts of Deep Learning in relation to this thesis are presented. Building on this, every step of the proposed approach is presented.

First, an introduction to the characteristics of the data sources used in this work and their acquisition is given (Chapter 2.1). This is followed by presenting general concepts of Machine Learning and increasingly specific Deep Learning concepts and techniques, such as Neural Networks, Convolutional Neural Networks and Fully Convolutional Networks (Chapter 2.2). At the end, the proposed approach is presented, including considerations derived from related studies and data sources (Chapter 2.3).

In a nutshell, the suggested approach can be described as follows: First, relevant OSM polygons and S2 images are obtained for the same areas. Afterwards, OSM data is classified by using a legend similar to the Corine Land Cover. Furthermore, a cloud detection algorithm facilitates the inclusion of an additional cloud class during several preprocessing steps. Annotation images present the basis for the training process of a Deep Learning classifier. After preparing and splitting patches of annotation and corresponding S2 images into training, validation and test datasets, a Fully Convolutional Network is employed for training and testing. The pretrained network is then utilized to train a second Fully Convolutional Network with a selected subset of the original dataset. This classifier is applied to predict S2 images for the chosen study area. Finally, the resulting LULC map is subjected to an accuracy assessment, where (class-wise) classification accuracy measures are determined. In addition, a VGI reference dataset is employed to provide additional insights about the classification (Figure 1).

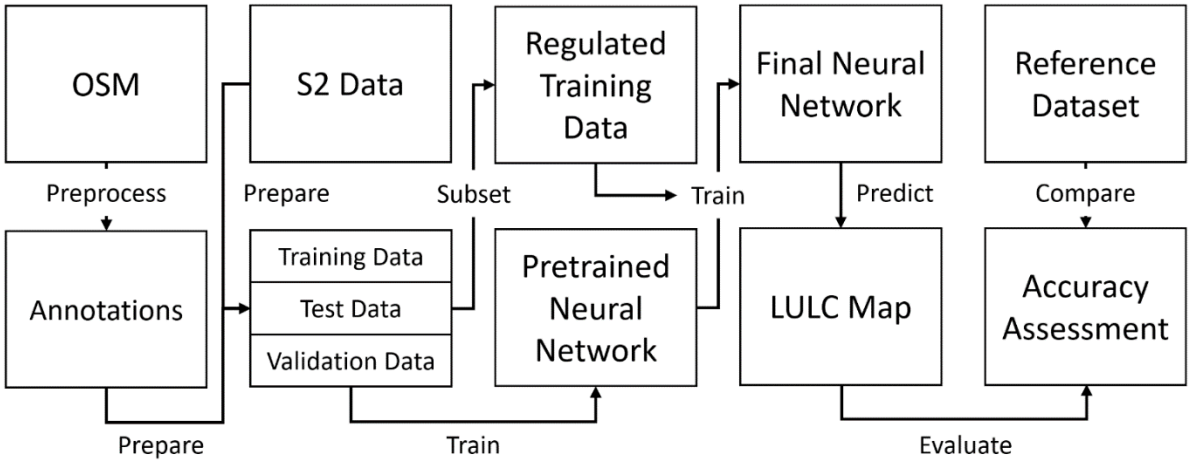

Figure 1: Diagram of the proposed workflow.

2.1 Data

This chapter describes datasets used to train, validate and test the Fully Convolutional Network (FCN) classifier. All external datasets are freely available and easily obtainable to ensure reproducibility and future improvement of the presented approach.

2.1.1 OpenStreetMap

Established in 2004, the OpenStreetMap (OSM) project today presents the biggest, most complete open-source geodatabase in the world. It is voluntarily built and updated by more than 5 million registered users (Wiki, 2019a) making it the largest Volunteered Geographical Information (VGI) project overall. Published under the Open Database License (ODbL), anyone can copy, share and alter OSM data, provided that OSM is cited appropriately and derived products are also published under the ODbL license. Contributors to OSM can add any spatial and thematic content they want - from a single bench to whole forests. Together with its free availability, this content diversity and abundance has made OSM an attractive data source for many use cases (Fonte et al., 2017).

OSM data are presented in a custom vector format, together with thematic information and comprises of nodes (point features), ways (line features) and relations (geometric collections, e.g. polygons). Thereby, there are no restrictions on the minimum mapping unit (MMU) for any application. Attributes for each geometrical feature are described through multiple tags. One tag consists of a key (general topic or type) and a value (specific form of the feature), which must be provided in pairs (e.g. highway=motorway). Although there is a list of established tags suggested by the OSM Wiki page (Wiki, 2019b), contributors are not restricted to any of those (Schultz et al., 2017).

When dealing with VGI information like OSM data, one central issue is always its quality. Just like the Wikipedia project, OSM relies on its contributors to provide complete, accurate and up-to-date information. The International Organization for Standardization (ISO) published a standard that defines the quality of geodata through the following five parameters: completeness, logical consistency, positional accuracy, temporal accuracy, and thematic accuracy (ISO, 2013). Many studies investigated OSM data towards those parameters for different use cases, highlighting its strengths and weaknesses.

OSM data may lack spatial and thematic coverage. This results in an uneven distribution of the data, leading to data gaps and missing thematic information (Neis and Zielstra, 2014; Zielstra and Zipf, 2010). Differences in positional and temporal accuracy were often found to be the largest in rural areas. Also, issues of spatially overlapping features may appear in OSM data, resulting in contradictory information (Schultz et al., 2017). So, when using OSM data, its heterogeneity in many respects must be considered. One specific task may only be feasible under certain circumstances for a restricted area.

Nevertheless, studies have shown great potential of OSM data for many applications, such as routing (Ludwig et al., 2011) and disaster management (Poiani et al., 2016). In addition, OSM data was successfully developed for LULC applications, contributing to the solution and mitigation of existing issues in that field (Estima and Painho, 2013; Fonte et al., 2016, 2017, 2019; Schultz et al., 2017).

2.1.2 Sentinel 2

Sentinel-2 is a remote sensing mission of the ESA (European Space Agency) set up to provide “global acquisitions of high-resolution multi-spectral imagery with a high revisit frequency” (Drusch et al., 2012). It was explicitly designed with free access to facilitate a generation of derivative products, such as landcover maps, change detection applications and geophysical measures. Two identical satellites (Sentinel-2 A and B) were launched in 2015 and 2017 and are successfully operating until today. The system provides multispectral data (13 bands) with a high spatial resolution (10m - 60m; depending on the band), a swath width of 290km, and a temporal resolution of five days (Drusch et al., 2012).

Seninel-2 data can be obtained freely at different processing levels, since it’s part of the Copernicus program (Sentinel Online - ESA, 2019). For this work only Sentinel-2 data of processing level 1-C is used, which was subjected to multiple preprocessing steps beforehand. These include radiometric and geometric corrections, resampling, ortho-rectification, image compression and the calculation of Top-Of-Atmosphere reflectances (Sentinel Online - ESA, 2019).

Deriving valuable information from this extensive data source continues to be an ongoing research topic. Existing applications for S2 data include forest and crop monitoring (Guo et al., 2018; Immitzer et al., 2016), biomass estimation (Sibanda et al., 2015) and slum mapping (Wurm et al., 2019), which confirms its high potential for land cover applications (Thanh Noi and Kappas, 2018).

2.1.3 Reference Datasets

Using reference datasets is common practice for LULC maps (Guo et al., 2018; Ndikumana et al., 2018; Nguyen et al., 2018). Therefore, a reference dataset was created for an area of 6.5km*6.5km to provide a complementary source of information for this approach, which helps assessing the suitability of LULC classes and underlying OSM features.

Reference data was systematically collected in the context of a VGI workshop between the Universities of Jena and Heidelberg from 11 to 12 July 2019. The workshop took place at the Institute of Geography at Friedrich Schiller University Jena under the lead of Dr. Michael Schultz, Benjamin Herfort and Janek Voß from the GIScience Research Group of the Heidelberg Institute for Geoinformation Technology (HeiGIT). It was organized and realized with the support of Dr. Chistian Thiel from the Department of Earth Observation and around 60 student volunteers. All participants were master students and future geography teachers from the University of Jena, who were provided with necessary skills beforehand. During the workshop, a continuous classification for this reference area could be obtained, using very high resolution (VHR) google satellite imagery (0.4m spatial resolution) from June 2019. Classes were hand-labelled in QGIS similar to the Corine Land Cover legend, following a labelling protocol (see Appendix). The resulting reference dataset is depicted in Figure 2.

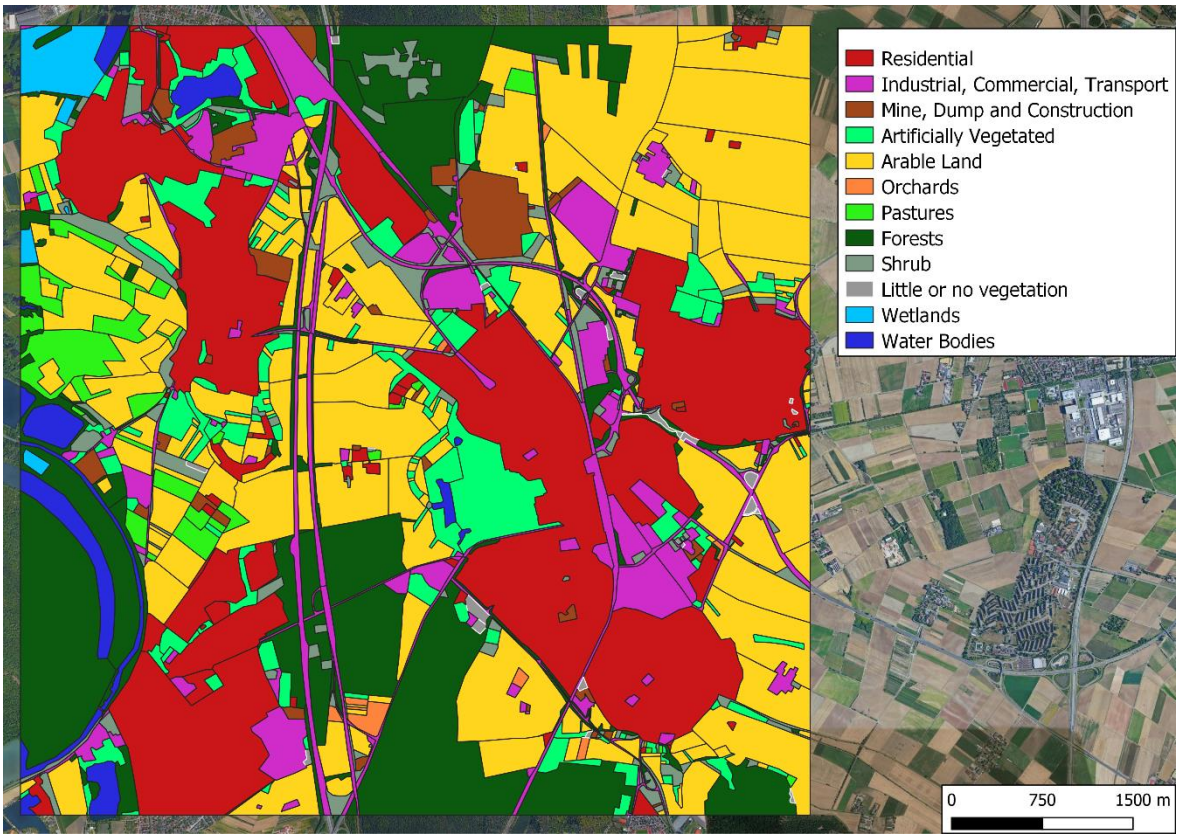

*Figure 2: Reference ground truth dataset (6.5*6.5km). (Abbreviated) legend derived from the Corine Land Cover legend.*

This area was chosen because of several factors. One strong reason was its great variety of LULC classes, shown in Table 1. In addition, OSM data density in this area was found to be above average, which facilitates a comparison to OSM data. It is also located within the boundaries of the study area of this work, to allow for comparison. Finally, preprocessed, cloud-free Sentinel-2 data could be obtained for the same spatial extent.

Table 1: Distribution of classes and area in hectare per class within the first reference dataset.

CLC Class	CLC Class Name	Area in ha	Class Proportion
1.1	Urban fabric	1048.78	24.83%
1.2	Industrial, commercial and transport units	374.60	8.87%
1.3	Mine, dump and construction sites	84.33	2.04%
1.4	Artificial non-agricultural vegetated areas	277.93	6.58%
2.1	Arable land	1301.78	30.82%
2.2	Permanent crops and orchards	12.70	0.32%
2.3	Pastures	98.21	2.32%
3.1	Forests	737.47	17.46%
3.2	Shrub and/or herbaceous vegetation associations	145.56	3.44%
3.3	Open spaces with little or no vegetation	13.93	0.32%
4.1	Inland wetlands	42.43	1%
5	Water bodies	85.99	2.03%
SUM		**4223.71**	**100%**

A second reference dataset created for this work is the result of an accuracy assessment performed to evaluate a LULC map (Chapter 2.3.6). This second database contains map class labels and their respective reference class labels for the same spatial locations. The number of labels, their spatial or class-wise distribution and the unit of assessment are dependent on sampling and response design of the accuracy assessment (Stehman and Foody, 2019). By organizing and quantifying this data, various accuracy measures of a LULC map can be calculated. This can include common accuracy metrics and a confusion matrix.

2.2 Deep Learning

Today, vast amounts of datasets are collected, distributed and stored every second and with increasing magnitudes (Witten et al., 2016). This technological development requires fast and reliable methods to extract meaningful information from those datasets. Among others, Machine Learning (ML) techniques like Deep Learning have found huge success and popularity recently (Zhu et al., 2017). In contrast to traditional data processing methods, ML methods are able to extract valuable information from data, without the need of specified instructions. Instead, predictions or decisions are learned by the algorithm itself from patterns in data (Witten et al., 2016).

Regarding their purpose, ML methods can be classified into different categories, such as supervised, semi-supervised, unsupervised and reinforcement learning. In this work a supervised learning approach is proposed due to the nature of the task. In supervised learning, already labelled data is used to train a ML model. This requires prior knowledge about the correct label of the respective training sample. From this, the algorithm can establish a relationship between input data and output label, using a function approximation.

Supervised learning may also be separated into classification and regression. In the context of classification, a ML model predicts a discrete output category (Y) for each input sample (X). Each prediction is expressed through a probability value (e.g. from 0.0 to 1.0), representing the likelihood of X belonging to Y. Depending on the application, output categories can be binary (Yes/No, True/False) or non-binary (Class A/B/C…). In contrast to that, regression models predict continuous numerical values as a function of input values (e.g. rental rates in response to city districts) (Goodfellow et al., 2016).

In the last decade, Deep Learning became an important research topic across disciplines, such as medical image analysis, recognition tasks and traffic flow prediction, outperforming existing approaches (Kussul et al., 2017; Ndikumana et al., 2018; Othman et al., 2016). Therefore, DL recently developed into a state-of-the-art method in remote sensing applications as well (Othman et al., 2016; Penatti et al., 2015; Zhu et al., 2017).

Using forms of Neural Networks, DL is a machines ability to automatically learn good feature representations from given data by combining simpler feature representation to obtain more complex ones (Goodfellow et al., 2016). Because DL methods recognize patterns in data by itself, they present a sub-category of Machine Learning. Both terms are part of the superordinate category Artificial Intelligence (AI). Whenever a machine is capable of solving a specific problem by using an algorithm, the term AI is utilized. However, for AI it is irrelevant which rule set, algorithm or technique is applied (Goodfellow et al., 2016). The relationship between the terms used is illustrated in Figure 3.

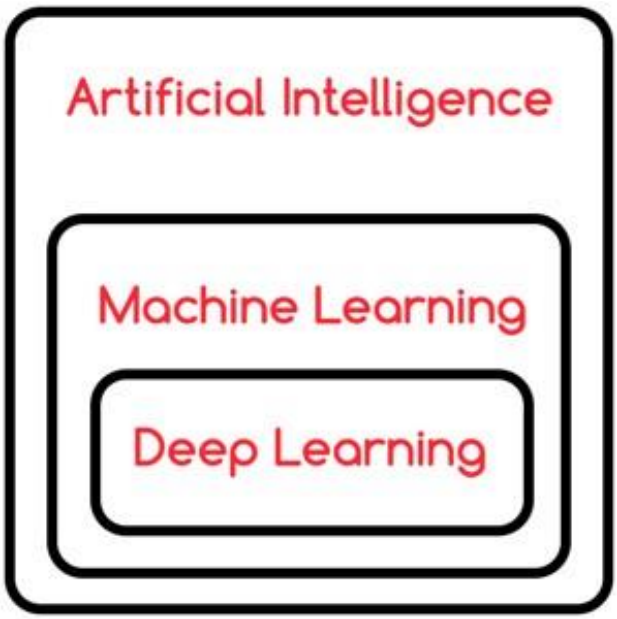

*Figure 3: Relation of the concepts Artificial Intelligence, Machine Learning and Deep Learning. Image downloaded from https://cdn-images-1.medium.com/max/1200/1*kz7IAKsfA80QROwXk10kdg.jpeg in March 2019*

Deep Learning concepts and techniques date back the 1940s. Since 2006 the term "Deep Learning" has become very popular in the field of Machine Learning (Chen et al., 2014). First works up to the 1960s dealt with biological learning and its application to ML, by developing artificial neurons. By 1986, the first Neural Network (NN) was developed and trained, using the training method backpropagation (Rumelhart et al., 1988). But until the last decade, NN could not be trained in an effective way, making DL a niche application.

This changed rapidly as Geoffrey Hinton et al. introduced an efficient training algorithm for NN in 2006 (Hinton et al., 2006). Together with the development of new software, hardware (especially GPUs), research finding and the presence of abundant datasets (keyword: big data), DL nowadays has become one of the leading technologies across many disciplines (Castelluccio et al., 2015; Goodfellow et al., 2016).

2.2.1 Neural Networks

A (feedforward) Neural Network (NN) is the main concept of DL. Its purpose is to approximate a non-linear function, which best maps input to output values. In the process, this model can learn a set of parameters that result in the best function approximation for given training data. Based on networks in the brain, biological findings were the source of inspiration to components of artificial NN. One central term borrowed from neuroscience is “neuron”. In a NN the neuron constitutes a mathematical unit, which can store, transform and pass information it receives to subsequent neurons. When information is processed downstream, without any feedback connections, the model is called a Feedforward Neural Network (Goodfellow et al., 2016).

The architecture of the feedforward Neural Network forms the basis for more sophisticated networks and consist of several neurons, organized in layers, and their connections. The number of consecutive layers in a NN is referred to as its depth. A Neural Network has an input layer, one or many hidden layers and a single output layer. Each layer can have a different number of neurons, which are fully connected to neurons in the adjacent layer (Figure 4).

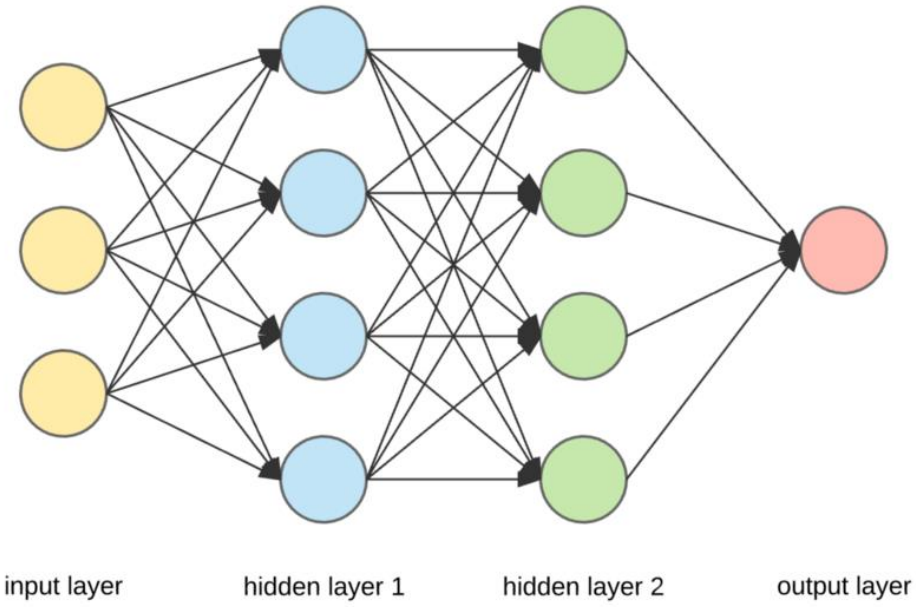

*Figure 4: Topology of a basic feedforward Neural Network. Image downloaded from https://cdn-images-1.medium.com/max/800/1*Gh5PS4R_A5drl5ebd_gNrg@2x.png in March 2019.*

Each neuron takes numerical values from all previous neurons and assigns weights to each of them, expressing their importance. Then the weighted sum over all input values is calculated and passed through a (non-linear) activation function. Finally, the result of this computation plus a bias value is used as an input to all neurons in the next layer, where those steps are repeated. The final output of a supervised NN can either be one neuron, containing a single numerical value (regression tasks) or a set of neurons (classification tasks), each holding a value. In a classification model, there are as many output neurons as there are classes. Thereby, the value of each neuron expresses the class probability of the initial input. To quantify the difference (error rate) between

calculated output values and the true value of the data, a cost function is used. Essentially, this cost function indicates how well the network performed for the current training sample (Patterson and Gibson, 2017).

Based on the computed cost, a NN can learn to minimize its error rate through an iterative learning process using the gradient descent algorithm. To describe how this algorithm works, we can use an analogy, where the function of all parameters (weights) within a model becomes a 3-dimensional landscape. Hills in this landscape represent parameter combinations resulting in higher costs (error rates), whereas valleys symbolize combinations of lower costs. To reach the bottom of a valley from any initial location, gradient descent takes steps towards the steepest direction (negative gradient). This process will repeatedly tweak the function (weights), measure resulting costs, and select new weight values that result in lower costs until a local or global minimum of the cost function has been reached (convergence) (Figure 5). The step size of the gradient descent is referred to as learning rate and presents one of the most important parameters of the model. If the learning rate is set too high, the algorithm might overshoot the minimum. If, however, the learning rate is too small, training would take too long and the algorithm can get stuck in small local minima (Patterson and Gibson, 2017).

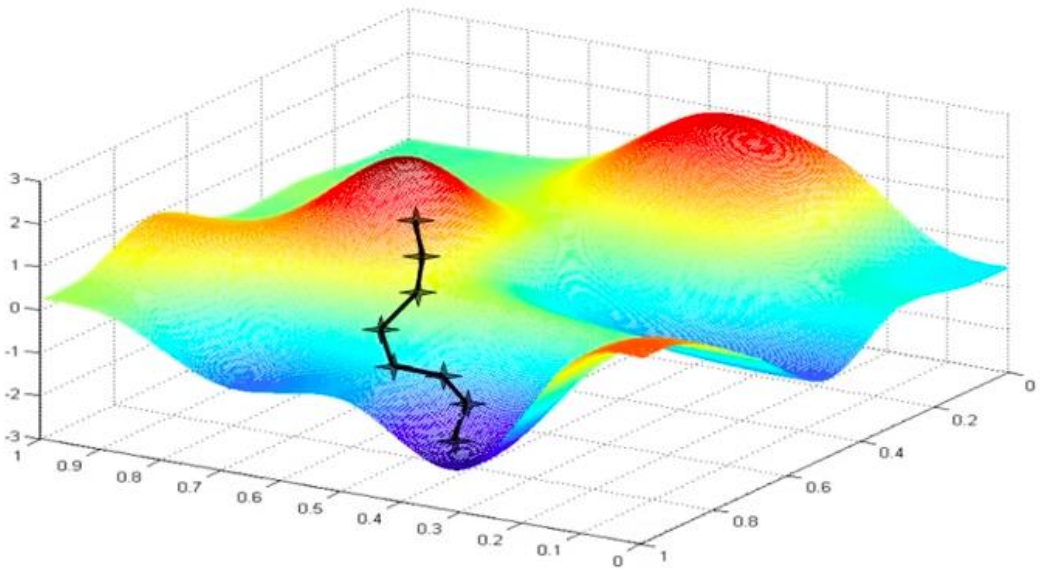

Figure 5: Example use of the gradient descent algorithm to reduce the cost of a function. Image downloaded from: https://i.stack.imgur.com/w7ARo.png in March 2019

To be able to apply gradient decent in a multilayer NN, a backpropagation algorithm is required. It updates connections between hidden layers using the gradient decent described before. Hereby, the algorithm works backwards, contrary to the network's processing direction. For each training sample, the backpropagation algorithm starts at the output layer and reversely iterates through all neurons of the hidden layers until it reaches the input layer. On its way, it computes how values should be changed for the most effective error rate decrease of the current training sample. Averaging all weight changes over all training samples gives the negative gradient for the cost function of our NN; in other words: the "direction" of change. However, it can become computationally very expensive to calculate a gradient descent over all training samples. So, in

practice, only a small random portion, called batch, of all training samples is used to calculate the gradient decent (Patterson and Gibson, 2017). Still, all DL networks take a long time to train, but are very fast once they are (Chen et al., 2014). When training a DL model, doing one forward and one backward pass over all training data is called an epoch. Increasing the size of the training dataset or the number of epochs, will increase the accuracy of the model, but especially in the domain of DL, there is always the risk of overfitting it (Goodfellow et al., 2016).

In ML models issues of overfitting happen when the model performs significantly better on training data, than on unseen (validation or test) data. This effect is triggered when the model simply memorizes the training dataset, instead of extracting meaningful features from it (generalization). With regularization the aim is to increase the performance of a model on validation and test data, even at the expense of increased training error (Goodfellow et al., 2016). In DL, there are many strategies to add regularization to a model. These include:

- Dropout:

  Dropout is by far the most used regularization technique in DL. The idea is to randomly turn off neurons and their connections with a predefined probability during the training phase of the network. At each iteration, dropout is initialized randomly again, so that each iteration a different set of neurons is active. This prevents the neurons from specializing too much, making the model more robust and general, but also reducing its capacity (Dertat, 2017).

- Dataset Augmentation/Increase:

  At stated before, more data generally results in a better performance. If there was an infinite amount of (correct) training data, overfitting would not happen because the model would know every instance of the data. Data Augmentation is a way to artificially generate more training samples by transforming existing ones. These synthetic samples can be created using techniques like rotation, shifting, resizing, exposure adjustment, contrast change and many more, resulting in a much larger training dataset (Dertat, 2017).

However, a DL model does not always have to be trained from scratch. Transfer learning uses an already trained model and repurposes it for another task. This method of fine-tuning a pretrained model is successful and effective, if both tasks and training datasets closely relate to each other (Castelluccio et al., 2015). If these conditions are met, transfer learning allows using less training data and decreases training time compared to training a DL model from scratch (Marmanis et al., 2016). In addition, it can help increasing the granularity of a classification (Wurm et al., 2019).

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) arose in the 1970s and describe a kind of Neural Network, which specializes in dealing with grid-like information, such as time-series or image data. Historically, CNNs developed out of neuroscientific findings by David Hubel and Torsten Wiesel in the 1960s. During their experiments, Hubel and Wiesel recorded the activity of individual neurons from the visual cortex of cats, while showing them different pictures. In the cat's brain neurons seemed to resemble patterns displayed on the respective picture in front it. They could also observe a division of tasks between neurons. Neurons in the early visual system most strongly responded to light patterns and edges within a specific area called the receptive field. In contrast to that, neurons in the later visual system behaved more invariant to changes in brightness, patterns and position of the picture shown. To summarize, it seems that within the visual cortex information is processed in a spatial way with an increasing level of abstraction (Goodfellow et al., 2016).

CNNs try to mimic this neuronal structure using a set of so-called convolutional layers. A convolution is a mathematical operation where at least two functions are offset against each other, resulting in a combination both functions. Transferred to a CNN processing an image, the image presents a two-dimensional function (x*y pixels), often called input. The input is convolved across defined pixel dimensions with a local function called a kernel, in other words a moving window function similar to the concepts of local filters (e.g. high pass and Soebel operator). In a CNN, the kernel slides over the input and computes a new value at every position. Afterwards, the output values of this process are stored in neurons, which form a so-called feature map (Figure 6).

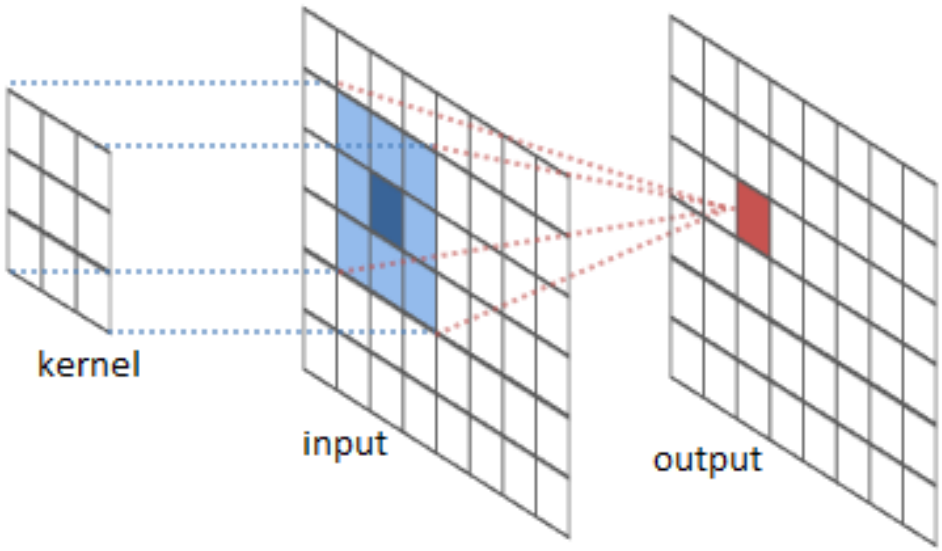

Figure 6: Visualisation of the convolving process within a CNN. The output can also be called feature map. Image downloaded from http://intellabs.github.io/RiverTrail/tutorial/images/convolution2.png in March 2016.

Since kernel dimensions are much smaller (e.g 3x3 pixels) than dimensions of the input data, not every input value is included in the computation of each output value, like it is the case in a feedforward Neural Network.

This property of CNNs called sparse connectivity significantly reduces memory requirements and improves computational performance of the model (Goodfellow et al., 2016).

The fact that the kernel does not change during the sliding operation is a form of parameter sharing and further reduces memory requirements of a CNN model. It will do the same operation over the entire input, always producing the same output (feature map). This is called equivariance of input and output. Due to the described structure, the kernel possesses a key role in any CNN to produce the desired output. Parameters of the kernel should be tuned and optimized, which iteratively happens during the training stage of the whole network. Besides the activation function, size, padding and stride are the most important kernel parameters.

An activation function is applied to all input values inside the kernel window to determine a single output value (activation) of one neuron in the resulting feature map. Influential to this output value are also size and padding parameters. Size specifies the size of the kernel (e.g. 5x5 or 3x3), whereas padding can be applied at the edge of the input (image). Here, where the kernel extends beyond the input, a padding preserves input dimensions and considers values at the edge (Figure 7). Lastly, the stride parameter defines the step size when moving the kernel over the input and thus influences the size of the resulting feature map (Dertat, 2017; Goodfellow et al., 2016).

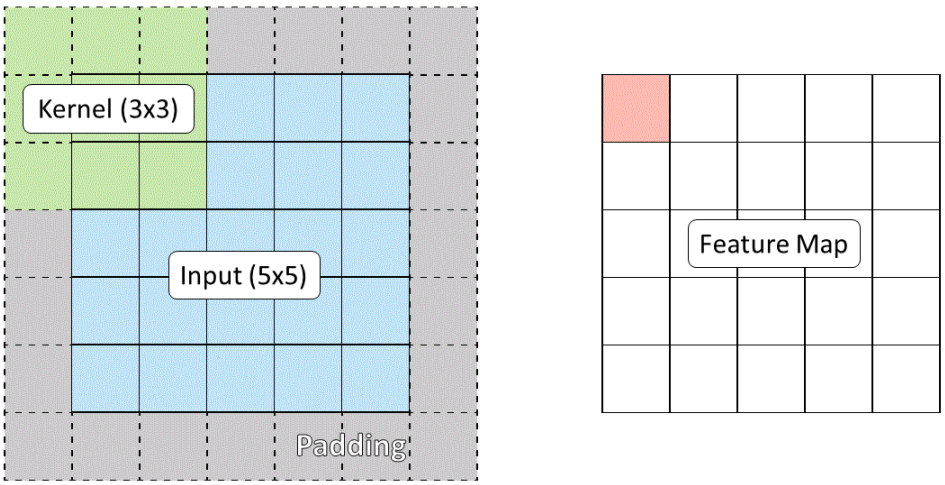

*Figure 7: Use of padding and step size of 1 to maintain dimensionality during a convolution. Own figure, based on image downloaded from https://cdn-images-1.medium.com/max/1200/1*W2D564Gkad9lj3_6t9I2PA@2x.gif in March 2016.*

The convolution process is applied to each layer (band) of the input image. The number of layers is referred to as depth of the input data. For an RGB-image for example, three convolutions (one for each band) produce three separate feature maps. In a CNN, convolution processes are chained after another, whereby the outputs (feature maps) of one convolutional layer act as input to a subsequent convolutional layer. Similar to the neurons in a cat's brain, the level of abstraction increases in the process as more complex features are calculated. The first feature maps extract low-level features from the input (e.g. edges from an image), whereas later feature maps automatically construct higher-level ones (e.g. circular spots in an image) from their predecessors (Castelluccio et al., 2015; Gao et al., 2018; Lavreniuk, 2017).

Pooling is a mathematical regulation of a CNN. It is usually applied between convolutional layers as a chained generalization process. By statistically summarizing nearby values (e.g. neighbouring values) from a feature map, pooling serves two purposes: First, it reduces the amount of values that must be stored within the network, therefore decreasing training time and computational costs. Second, pooling makes a CNN invariant to small changes of the input, similar to the neuronal processes observed in a cat's visual cortex. Rather than preserving the exact position of a feature, pooling emphasizes its existence and its rough location relative to other features (Dertat, 2017; Goodfellow et al., 2016).

Pooling type, pooling window and stride are mandatory parameters of CNN pooling operations. The pooling type specifies the kind of pooling applied to a feature map. The most common one is max-pooling, which only preserves the maximum value present inside the pooling window (Goodfellow et al., 2016; Volpi and Tuia, 2017). The pooling window with a definable size (e.g. 2x2 pixels) in turn slides over the feature map with a specified stride (e.g. 1 pixel) resulting in a smaller so-called pooling layer (Figure 8).

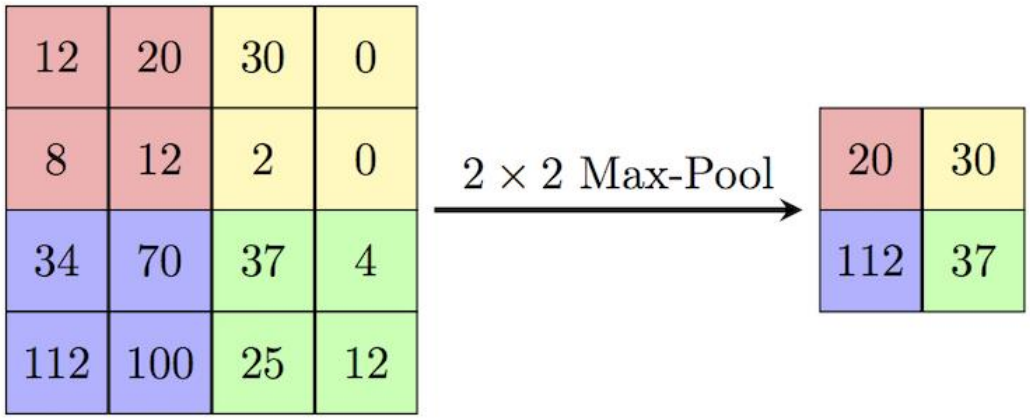

*Figure 8: Application of max pooling with a pooling window of 2*2 and a stride of 2 to generalize a convolutional layer into a pooling layer. Image downloaded from https://computersciencewiki.org/images/8/8a/MaxpoolSample2.png in March 2016.*

Putting convolutional layers and pooling layers together constitutes for large part of a CNN. This step can be described as the feature extraction part of the CNN. The last step is to classify the previously extracted features. This is done by flattening the values from the last layer into a one-dimensional array and passing it to one or more fully connected layers (FC-layers), which are built exactly like layers in a Neural Network. Finally, the output in the form of, for example, class probabilities is obtained using an activation function like Softmax, Rectified Linear Unit (ReLU) or Sigmoid in the final FC-layer (Goodfellow et al., 2016) (Figure 9).

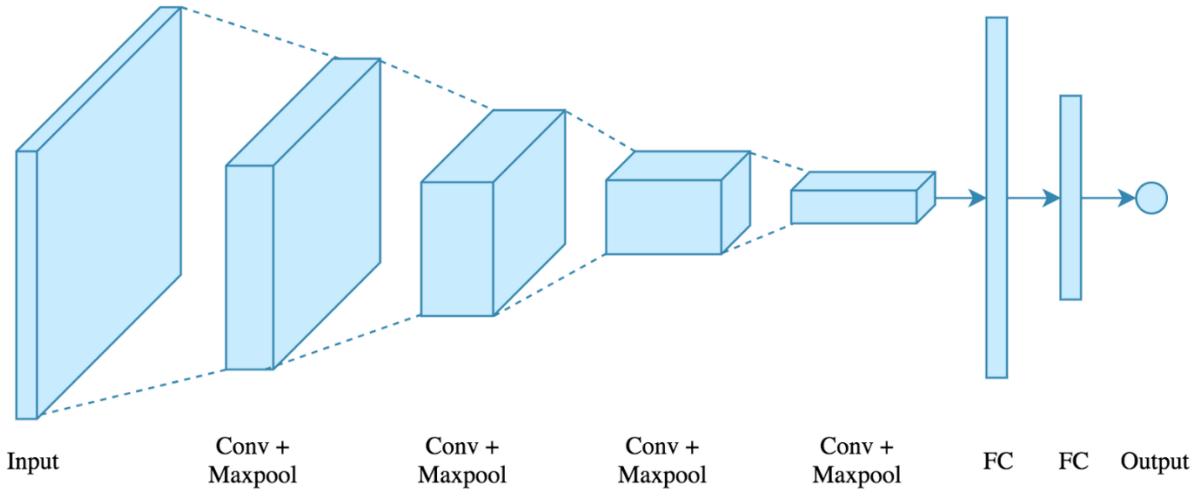

*Figure 9: Schematic Illustration of the general structure of a CNN. Image downloaded from https://cdn-images-1.medium.com/max/1200/1*uUYc126RU4mnTWwckEbctw@2x.png in March 2019.*

In recent years, Convolutional Neural Networks among other Deep Learning techniques have become one of the leading techniques, most commonly used for image analysis. As a result, they have become the gold standard for many image-related tasks (Volpi and Tuia, 2017). The main factor for this development was their success in multiple image applications (Goodfellow et al., 2016; Volpi and Tuia, 2017). In the remote sensing area, CNNs have proven to be effective in a variety of tasks, such as scene classification, (hyperspectral) image classification and semantic segmentation (Zhu et al., 2017).

2.2.3 Fully Convolutional Networks

In Semantic Segmentation (pixel-wise classification) each pixel of an input image is classified individually. When applying CNNs for Semantic Segmentation tasks, the class of each pixel is generally determined using features extracted from the enclosing region (kernel window), disregarding information beyond the edge of it. Also, spatial information about extracted features is inevitably lost when using FC-layers at the end (Long et al., 2015). By increasing kernel size or using pixel patches, deeper and broader features can be extracted by the CNN with the disadvantage of increased computational costs and higher loss of spatial accuracy. To resolve this tension between information and location, Fully Convolutional Networks (FCN) as a variant of CNNs were introduced in 2015 (Long et al., 2015).

FCNs preserve the 2-dimensional structure of the input image by replacing FC-layers in a CNN with convolutional layers. In an FCN, down-sampling through convolutional and pooling layers is complimented with up-sampling/deconvolutional layers. Deconvolutional layers transpose the information from the last convolutional layer of a CNN to an output layer by utilizing learnable up-sampling parameters. This extrapolation is carried out over one or more deconvolutional layers (Long et al., 2015; Shibuya, 2017). The result is a reconstructed input image predicting class labels pixel-wise (Figure 10).

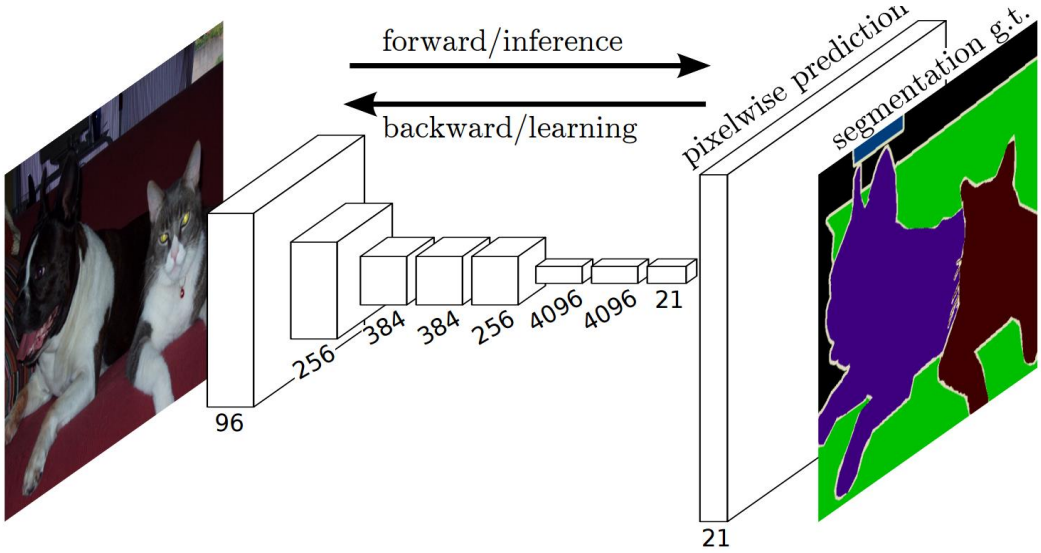

Figure 10: Structure of the FCN for Semantic Segmentation used by Long et al., 2015. Multiple down-sampling layers are following by one up-sampling layer. Image downloaded from http://www.deeplearning.net/tutorial/_images/cat_segmentation.png in April 2019.

Due to its design, FCN offer several advantages. First, it can use an arbitrary sized training dataset to produce respectively re-sized output classifications. Second, FCN obtain higher accuracy rates for semantic segmentation tasks, since they consider a larger area of the image for feature extraction. And thirdly, computational costs are reduced, since each pixel value is only considered once per input image (Badrinarayanan et al., 2017; Fu et al., 2017).

A central problem when using a simple convolutional-deconvolutional FCN remains: The output image becomes coarse and boundaries blur, since the last convolutional layer of a CNN only presents a fraction of the input dimensions (width, height). Thus, state-of-the-art FCNs use multiscale classification techniques, such as skip-layer network architectures. Here, the output from layers at different levels of the down-sampling part are used as inputs for deconvolutional and classification layers. This results in multiple classifications at multiple resolutions for the same image. After a bilinear interpolation, which rescales resulting layers to the starting resolution, all classifications can be combined to the final output classification using a FC-Layer (Fu et al., 2017; Long et al., 2015) (Figure 11).

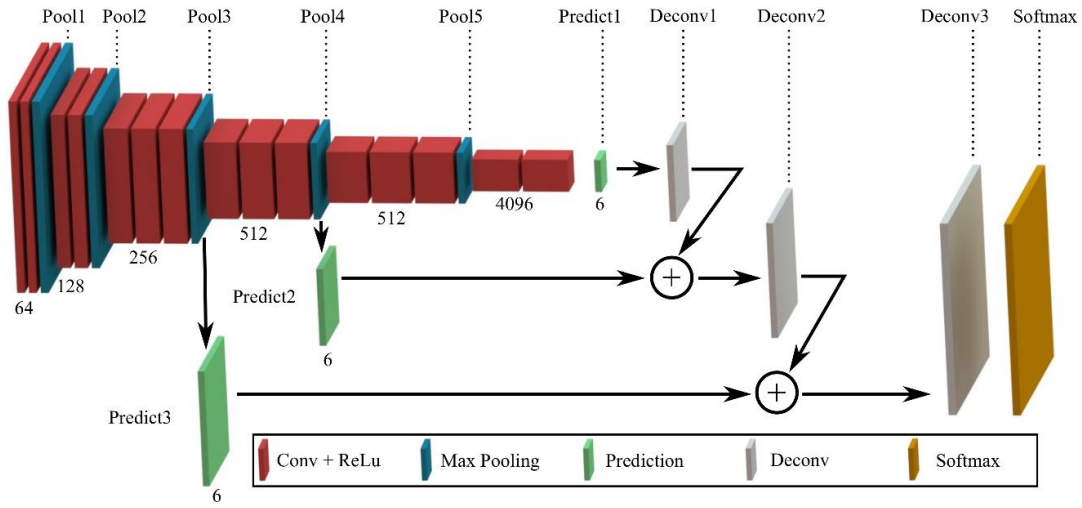

Figure 11: Exemplary illustration of an FCN with a skip-layer architecture combining three deconvolutional layers (prediction layers). Image downloaded from https://blog.playment.io/wp-content/uploads/2018/02/fcn_arch_vgg16.png in April 2019.

New FCN models and their predecessors are constantly evolving in the field of DL. A prominent representative of this model family is the U-Net model, initially designed for biomedical image segmentation (Ronneberger et al., 2015). However, since its development in 2015 it was applied for other segmentation tasks as well, winning several Kaggle competitions in the fields of Image Masking, Seismic Image Segmentation and Satellite Image Segmentation (Kaggle Team, 2017; Lamba, 2019; Nguyen et al., 2018).

The U-Net is built upon a standard multiscale FCN structure extending the up-sampling part of the network with additional feature channels. The result is a symmetrical, U-shaped fully convolutional network with multiple skip-layer structures and 23 convolutional layers (Figure 12) (Ronneberger et al., 2015).

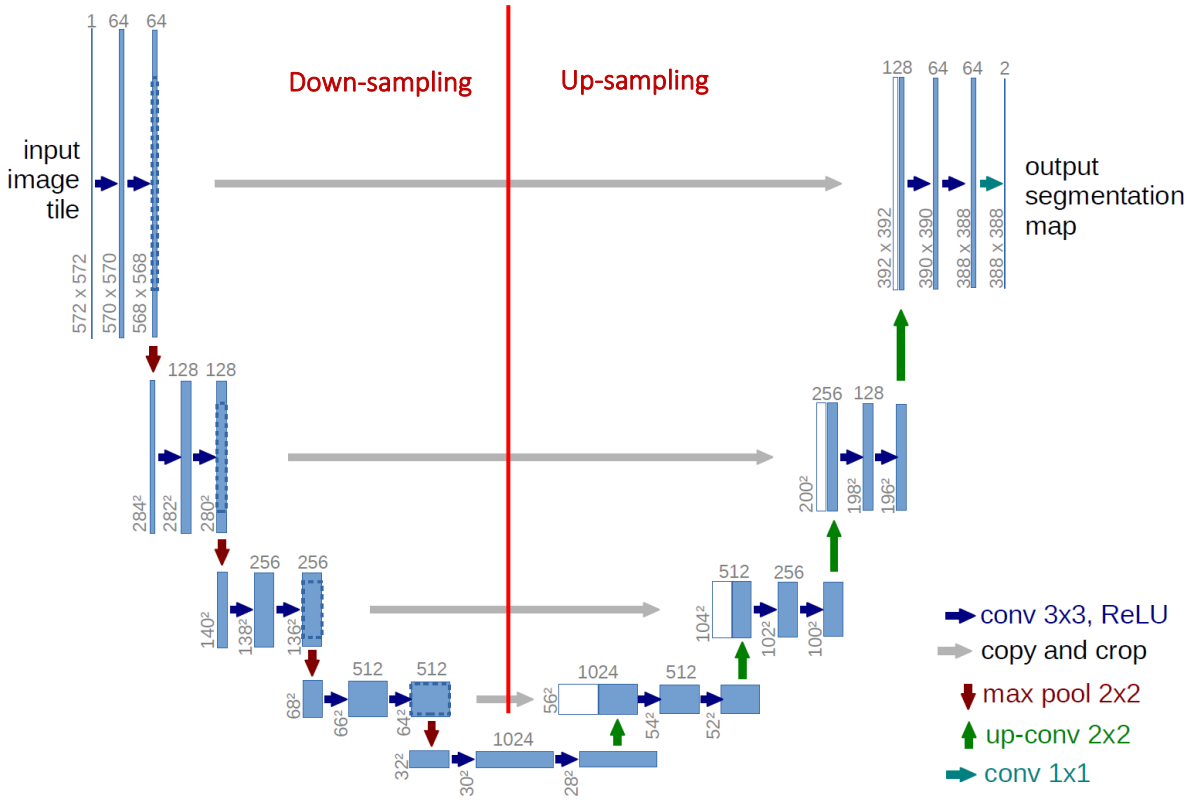

*Figure 12: Network architecture of the U-Net with its Down- and Up-sampling parts entirely consisting of Convolutional Layers. Note that initial image dimensions (572*572 pixels) are interchangeable. Figure based on Ronneberger et al., 2015.*

2.3 Approach

Based on the methods presented in the previous section, the approach used in this work was developed. The first step is to make general considerations about the task, which involves requirements against subsequent steps of the workflow. Those then consist of multiple preparation measures, namely Data Acquisition, Preprocessing and Training Preparation (Figure 13). Eventually the data thus generated is used to train the DL classifier. Here, parameter choices and training iterations, as well as the architecture of the network is taken into account. The trained classifier is lastly evaluated using an accuracy assessment.

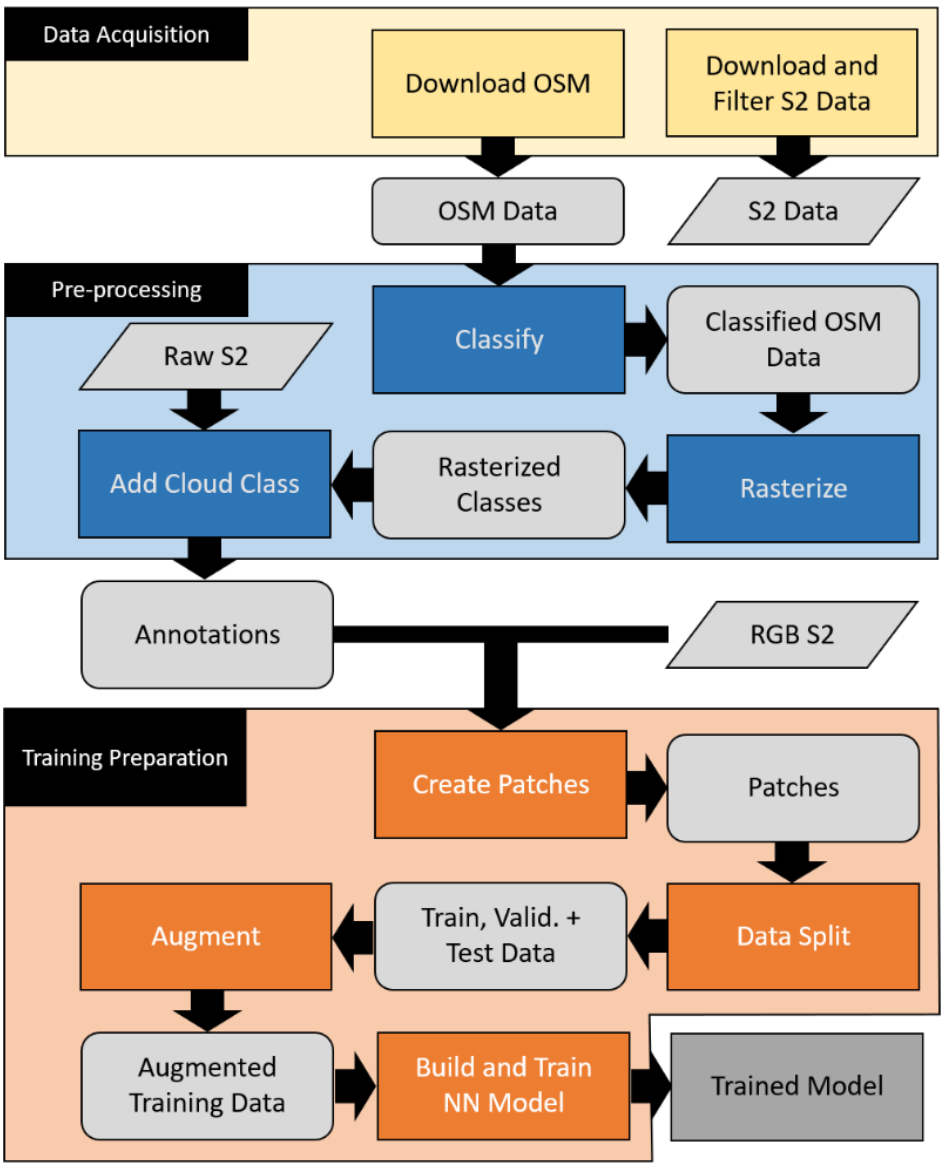

Figure 13: Depiction of preparation steps, divided into Data Acquisition, Preprocessing and Training Preparation. Colours are for aesthetic purpose only.

2.3.1 General Considerations

Classifying remote sensing data over large territories and extended time periods comes with challenges, since LULC features greatly vary in terms of climatic, topographic and geobotanical conditions (Henry et al., 2019; Morrison and Olson, 2005). Over centuries, biogeographical concepts were developed continuously to be able to summarize regions of comparable flora and fauna. One project, the Digital Map of European Ecological Regions (DMEER), developed by the European Environmental Agency (EEA) and the World Wildlife Fund (WWF), introduces the concept of ecoregions covering Europe (Morrison and Olson, 2005). Ecoregions are "nested within biogeographic realms and biomes", sharing the same external borders (Olson et al., 2001). They evolved from both historical and regional classification systems as a collaborative effort from more than 1000 experts of various disciplines. Despite their distinct boundaries, ecoregions should be used with caution, because they present a compromise between different taxa and can only present an imperfect abstraction of reality (Olson et al., 2001).

The study area of this work extends across one specific ecoregion in Europe, the "Western European broadleaf forests", due to its abundance of OSM data (Wiki, 2019a) (Figure 14). The study area has a size of approximately 492.329km². While the choice of the study area is arbitrary using this approach, the step of restricting it to an ecoregion likely facilitates a more characteristic and distinct feature space, which makes classification tasks applied to it more straightforward. Also, any chosen study area should contain enough land use related OSM data, because it's used to generate training data for our approach (see Figure 1).

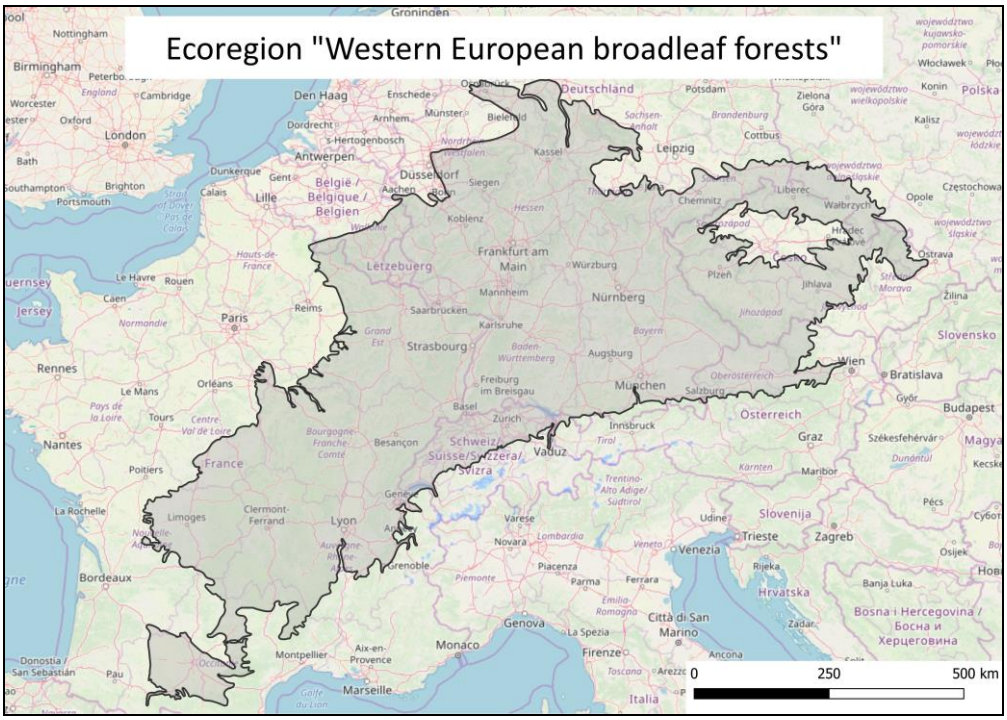

Figure 14: Depiction of the study area, which comprises of the ecoregion "Western European broadleaf forests" developed by EEA and WWF.

The effect of seasonality highly influences land use and land cover, especially when it comes to agricultural land, broadleaf forests, pastures and mountain regions (Kussul et al., 2017). Similar to the choice of a study area, restricting the acquisition period of all data to one specific season will most likely lead to a more concise and precise classification. For this approach, the meteorological summer season of the northern hemisphere in 2018 (2018/06/01 – 2018/08/31) was chosen, because of its reduced cloud cover on average. All satellite images were acquired within this time period and a snapshot of OSM data from the last day of summer 2018 (2018/08/31) was taken.

Lastly, the percentage of cloud cover within remote sensing data is considered. Only satellite images with less than 20% cloud cover are selected. Clouds will always cover underlying classes and thus reduce the amount of data available for training (Kussul et al., 2017; Ndikumana et al., 2018). Furthermore, context information between neighbouring classes can be considered and learned as high-level features by a DL classifier without any cloud cover (Fu et al., 2017). By using images with less cloud cover, spatial relations between classes are more likely to be preserved, meaning the FCN is able to learn features from them (see Chapter 2.2.1 and 2.2.2).

2.3.2 Data Acquisition

Within the proposed workflow (Figure 1), the first step is to obtain necessary data, more specifically, LULC-related OSM and S2 data. With regard to considerations defined in Chapter 2.3.1, season, cloud cover, spatial and temporal resolution should be paid attention to when acquiring RS data. In order to suceed, the DL classifier also has specific requirements against size and format of training data. Therefore, the study area was divided into approximately 10500 tiles with an extent of 6.5km*6.5km (Figure 15). At last, OSM and S2 data with corresponding spatial extents must be associated to each other unambiguously (e.g. by naming them similarly).

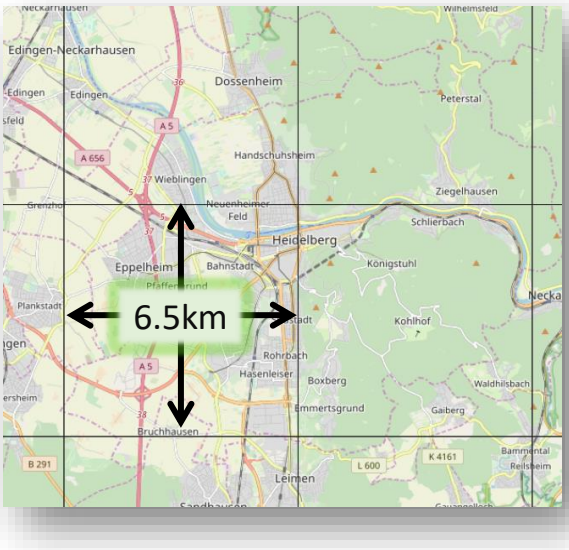

*Figure 15: Extract of the study area delineating multiple tiles (6.5km*6.5km) used to create the training data set.*

Already filtering OSM data in this step reduces both dataset size and acquisition time. Since LULC related OSM features are used in this approach, only tags with the keys “landuse”, “natural”, “leisure”, ”tourism” and “waterways” need to be obtained. This selection of keys is based on the work of Schultz et al., 2017.

To fulfil those requirements “sentinelsat Python API” and “Ohsome REST API” along with basic Python libraries were used. Those APIs facilitate the use of necessary parameter options to be able to create the desired training dataset and are free to use. As a result, both OSM and S2 data is obtained.

2.3.3 Preprocessing

Preprocessing covers the process of transforming raw OSM vector data (relations and ways) into annotation raster images used to train the DL classifier (see Figure 13).

Initially, OSM data is further filtered and transformed, creating landuse and landcover classes. Excluding incomplete images from all following steps was necessary, since some satellite images obtained for the study area could not fulfil the requirement of less than 20% cloud cover within the summer season of 2018. Others would not cover the entire extent of a tile (6.5km*6.5km), because trajectories of S2 satellites are not always congruent to the extent of tiles. This data loss however must be expected given the relatively short time interval of summer 2018 and a 5-day temporal resolution of S2 images. Therefore, approximately 9200 images, out of potentially 10500 images (one for each tile) were used for subsequent steps.

For the legend harmonization process all non-polygonal OSM features were discarded. Remaining LULC-related polygons are now attributed according to their tag values (Table 2), which results in classified OSM polygons. Assignment of tag values to classes follows the approach presented in Schultz et al., 2017.

Table 2: Legend harmonization between OSM tags and Corine Land Cover (CLC) classes, level two legend.

CLC Class	CLC Class Name	Corresponding OSM tag values
1.1	Urban fabric	residential
1.2	Industrial, commercial and transport units	industrial, commercial, retail, harbour, port, railway, lock, marina
1.3	Mine, dump and construction sites	quarry, construction, landfill, brownfield
1.4	Artificial non-agricultural vegetated areas	stadium, recreation_ground, golf_course, sports_center, common, allotments, playground, pitch, village_green, cemetery, park, zoo, track, garden
2.1	Arable land	greenhose_horticulture, greenhouse, farmland, farm, farmyard
2.2	Permanent crops and orchards	vineyard, orchard
2.3	Pastures	meadow
3.1	Forests	forest, wood
3.2	Shrub and/or herbaceous vegetation associations	grass, greenfield, scrub, heath, grassland
3.3	Open spaces with little or no vegetation	fell, sand, scree, beach, mud, glacier, rock, cliff
4.1	Inland wetlands	march, wetland
5	Water bodies	water, riverbank, reservoir, basin, dock, canal, pond

Classified OSM polygons were rasterized in the next step (see Figure 13). Here, it was ensured that rasterized OSM classes showed similar spatial resolution, projection and extent regarding corresponding S2 images. The common issue of overlapping OSM data leading to ambiguous information was handled by always preserving smaller polygons respectively (Schultz et al., 2017).

The last step towards creating ground truth raster data was the detection and classification of possible clouds in corresponding S2 images and their transfer to the respective OSM raster as an additional class (Figure 13). Because cloud cover conceals every underlying information in an S2 image, any existing classification in the related OSM class raster should be overridden with a cloud class at these locations to facilitate an effective training of the DL classifier.

Clouds were detected using Python library "s2cloudless". The algorithm uses all 10 bands of an unprocessed S2 image and assigns cloud probabilities for each of its pixels (Zupanc, 2017). It also allows averaging of probabilities over neighbouring pixels and dilation of the cloud mask. Therefore, those parameters offer a lot of flexibility in terms of reliability, help mitigate salt and pepper effects and assist in dealing with outlier values. In this work a cloud probability of 45%, an averaging of 2 neighbouring pixels and a dilation size of 4 pixels was used. Based on empirical experience, this parameter combination presented a compromise between reliability and detection sensitivity of clouds for data within the study area.

The result of all preprocessing steps is an OSM class raster image, which can be assigned explicitly to a corresponding S2 image and will be used as annotation data for the DL classifier (Figure 16).

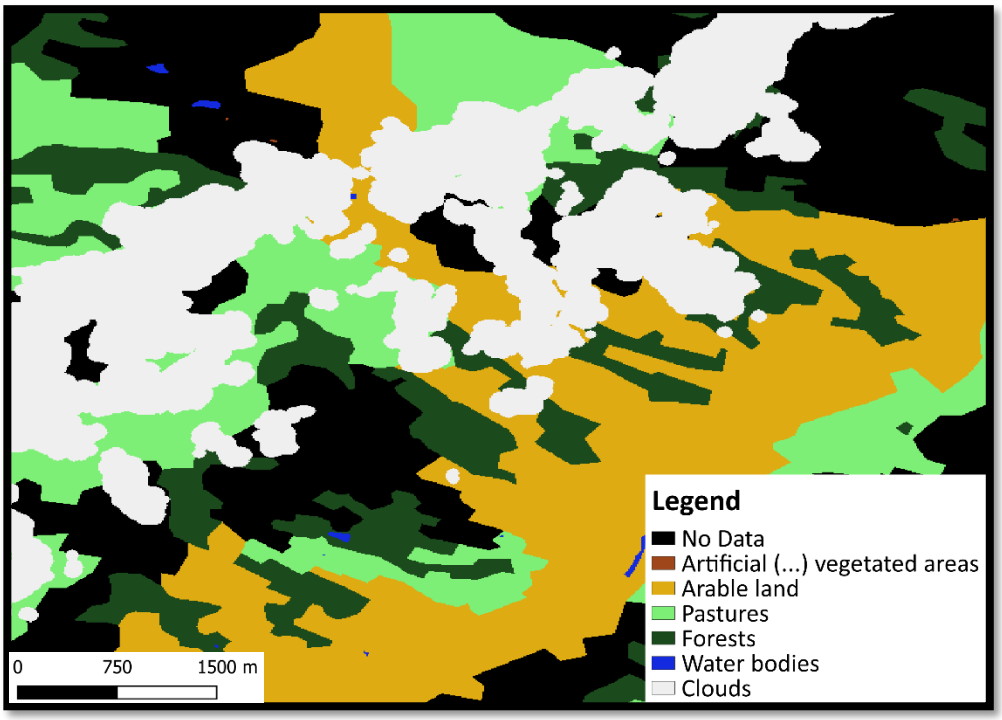

Figure 16: Example of an annotation image after the preprocessing of OSM data.

2.3.4 Training Preparation

Approximately 9200 data samples (annotations + corresponding S2 images) resulting from preprocessing steps (Chapter 2.3.3) were utilized for training, validation and testing of the DL classifier. One data sample consists of both S2 image and the corresponding annotation image.

Initially, coherent patches with a chosen width and height are extracted from randomly selected data samples (Figure 17). At the same time, the position of each extracted patch is randomly set within the bounds of the respective data sample. This allows for heterogenous image dimensions of data samples and produces any desired amount of training data, limited only by the memory capacity of the device used for training the model (Fu et al., 2017).

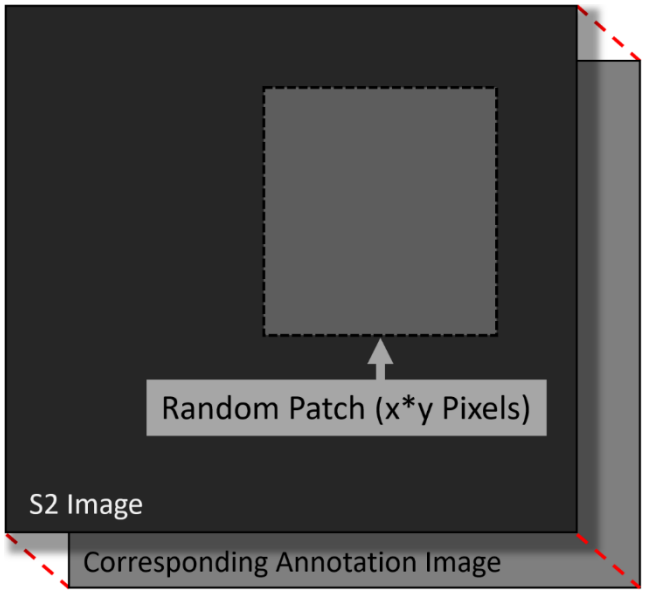

Figure 17: Example of one patch extraction used for training the DL classifier

With larger patch dimensions (x*y pixels), a DL network will be able to consider more and more complex features with the detriment of higher computational costs (Fu et al., 2017). Hence, both patch dimensions and number of patches should increase training time and performance of the classifier. However, the Python library Keras does not support ignoring a specific class for training a semantic segmentation model. Therefore, gaps in training data are always treated similar to LULC classes. During training, the model tries to find regularities and patterns for this gap class as well, so that under certain circumstances, pixels can be misclassified, which deteriorates the resulting prediction. To tackle this problem, every patch is checked for gaps (no data values) in the related annotation image. Only if a patch contains a certain amount of data, it's considered for augmentation and used for training later. The minimum data density used in this approach was set to 80%. Consequentially, all patches with a no data proportion of 20% or higher are disregarded hereafter. Patches are generated and loaded into the cache in this way until an arbitrary dataset size (number of patches) or the memory limit of the training device is reached.

Balance of under and overfitting is essential (Castelluccio et al., 2015; Dertat, 2017; Nguyen et al., 2018), hence existing data is augmented, multiplying its quantity. Choosing appropriate augmentation techniques is key and

depends on the performed task. A common practice for image data is to perform colour and/or geometric augmentations, such as rotation, flipping, transformation, channel shuffle, grayscale and contrast variation (Perez and Wang, 2017). However, for this approach only rotation, batch normalization and flipping methods are applied, similar to comparable studies (Castelluccio et al., 2015; Ioffe and Szegedy, 2015; Nguyen et al., 2018; Volpi and Tuia, 2017). Batch normalization is applied to combat variable illumination conditions between images, but additional colour augmentation methods can limit the model's capacity to identify LULC classes with distinct colour representations (Ndikumana et al., 2018; Volpi and Tuia, 2017). Also, any augmentations that deform a RS image could cancel out characteristic shapes, such as the linearity of rivers or the angularity of agricultural fields.

When setting up a Machine Learning model, it is common practice to use three different datasets, which are applied at different stages of the workflow (Shah, 2017). First, a training dataset is used to fit the model against given data by adjusting its parameters. Second, the trained model is evaluated on the validation dataset. Here, previously unseen data is facilitated to tune model hyperparameters. After alternating first and second step, a final, unbiased evaluation can be obtained by running the trained model on a separate test dataset to determine the final performance of the model (Shah, 2017).

For this approach, those three datasets are obtained by randomly splitting all patches into training, validation and test datasets with a rate of 6:2:2 (Chen et al., 2014). Aforementioned augmentation methods are then applied randomly and on-the-fly to training and validation datasets, while forwarding them bit by bit to the classifier.

2.3.5 Model Setup

The DL classifier U-Net model was chosen, due to its straightforward implementation, comprehensive documentation and proven success in the field of remote sensing (Kaggle Team, 2017; Lamba, 2019; Ronneberger et al., 2015). Model and parameters were implemented within Python using the DL library Keras running on top of the ML framework Tensorflow. An overview over essential model parameters used in this approach can be found in Table 3.

For training, a maximum of 50000 patches with the size of 256*256 pixels could be collected before the main memory capacity of the server was reached. With that data, the U-Net was repeatedly trained for 92 epochs with a batch size of 32 patches using Keras' Early Stopping method, which automatically terminates the learning process if no improvement of metrics within a defined number of epochs ("patience" parameter) occurs. The network learned using the "Adadelta" optimizer, since this learning rate method "dynamically adapts over time [...] and has minimal computational overhead beyond vanilla stochastic gradient descent. The method requires no manual tuning of a learning rate and appears robust to noisy gradient information, different model architecture choices, various data modalities and selection of hyperparameters" (Zeiler, 2012). Learning progress was measured using validation data in combination with the metric "Sparse Categorical Cross Entropy", which quantifies the distance between true and predicted label of each pixel ("Cross Entropy") across multiple classes ("Categorical"), which in turn are encoded as integers ("Sparse") (Goodfellow et al., 2016).

Table 3: Overview of parameters used for the training of the first DL-classier.

Parameter	Value
DL-Model	U-Net
Patch Size	256*256
Training Data Samples	30000
Validation Data Samples	10000
Test Data Samples	10000
Epochs	Early Stopping: Patience = 7 Epochs, Metrics = Sparse Categorical Accuracy
Loss Function	Sparse Categorical Cross Entropy
Evaluation Metrics	Sparse Categorical Accuracy + Loss
Learning Rate	Adadelta (adjusting Learning Rate)
Batch Size	32 (default)
Dropout Rate	25%

After the first training, 9200 data samples were revised, and a small subset of samples was selected for a second training. To this end, 30 data samples with high completeness, class existence and class proportions were manually chosen to fine-tune the initial classification using transfer learning (Chapter 2.2.1). A copy of the initial U-Net model was built with weights from the first training and afterwards trained with similar parameters to create an improved version of first classifier (Table 4). In this work, transfer learning can only be applied, because both training datasets are comparable, and classification tasks are identical. Previous studies underline the advantages of this method, since it was able to achieve improved classification performance with very little training data in similar use cases (Castelluccio et al., 2015; Marmanis et al., 2016; Wurm et al., 2019).

Table 4: Parameters used for the training of the second and final DL-classier. This table includes modified parameters only. Other parameters did not change between first and second training.

Parameter	Value
Training Data Samples	10000
Validation Data Samples	1000
Test Data Samples	1000
Batch Size	16

Both trainings were carried out on an Amazon Elastic Compute Cloud instance (EC2). The instance type used is called p2.8xlarge intended for general-purpose GPU compute applications. It includes 488 GB of RAM, 8 NVIDIA TESLA K80 GPUs (96 GB) and the Intel Xeon E5-2686 v4 (32 *2.3-3.0 GHz) virtual CPU processor. Training a DL network on GPUs allows for much faster training compared to training it on a CPU processor (Zhu et al., 2017)

2.3.6 Accuracy Assessment

During the automated evaluation process on the test dataset any prediction the model makes is compared to the class in underlying annotation data. Considering the particularities and challenges of OSM data, it becomes clear that this evaluation strategy is not sufficient for testing the model's performance. This is because underlying annotation data is based on OSM (+clouds), which does not present a reliable source of information (Chapter 2.1.1). To still be able to determine the quality of the classification, an accuracy assessment of a derived map is proposed. This map is generated by using the final model to predict LULC classes for collected S2 RGB-Images (approx. 9200) within the study area.

Accuracy assessment in the field of RS is a well-established and reliable instrument to evaluate the thematic accuracy of LU and LC products and is thus regarded as the gold standard (Stehman and Foody, 2019). The accuracy assessment quantifies the agreement between predicted and true classification of every pixel in a map, however, it does not cover other aspects of the quality of geodata (Chapter 2.1.1) (ISO, 2013). The three key components of an accuracy assessment are sampling design, response design and analysis (Strahler et al., 2006).

2.3.6.1 Sampling Design

"The sampling design is the protocol for selecting the subset of assessment units for which the reference classification is obtained and then compared to the map classification" (Stehman and Foody, 2019). Here, a common strategy of "standard random stratified sampling" is applied to randomly distribute an adequate number of assessment units (reference points) across all classes, respecting possible class imbalances. The number of assessment units for each class, called sample size, is calculated from formula 1 (Foody, 2009):

(1) $$n = \frac{z^2_{\propto/2}\, P(1-P)}{h^2}$$

Where n is the sample size for each class, h the confidence interval, P the respective class proportion of the classification and $z_{\propto/2}$ the critical value of the normal distribution, depending on the significance level α. Values were set following conventional practice with h = 0.05 and α = 0.95 so that $z_{\propto/2}$ = 1.96 (Foody, 2009; Schultz et al., 2017).

2.3.6.2 Response Design

"The response design defines how a decision on the agreement between the predicted (map) class label and the reference class label is made"(Stehman and Foody, 2019). This includes the spatial unit of the assessment, labelling protocol, background imagery and definition of thematic agreement.

For this approach, the spatial unit of assessment corresponded to S2 RGB pixels with a size of 10m. All reference points collected were randomly shuffled across all classes to prevent label regularities. At each reference point the class label occupying most of the pixel (>50%) was derived from very high resolution (VHR) data (bing aerial image) from late 2018 to minimize time discrepancy between map and accuracy assessment. In case clouds occurred within the scene, original S2 images were consulted to validate the cloud class. The label of each pixel was set by RS experts via visual interpretation. In the process, interpreters were unaware of the predicted map label and followed a labelling protocol. The contents of the labelling protocol included minimum mapping unit, which was set to 1 pixel, and class definitions specified in the official Corine nomenclature. Class definition used for the labelling protocol can be found in the appendix of this work.

2.3.6.3 Analysis

The third and last component of the accuracy assessment is the analysis. The focus here is on organizing and quantifying different accuracy measures derived from map and reference classifications. Traditionally, this is done by using a comprehensive error matrix (Stehman and Foody, 2019). Common metrics addressed within the error matrix are Overall accuracy (OA), Producer's accuracies (P_j) and User's accuracies (U_i). Producer's accuracies (P_j) are calculated class-wise as the ratio between correctly classified reference points (P_{jj}) and total reference class labels (P_{+j}) (Congalton, 1991):

(2) $$P_j = p_{jj}/p_{+j}$$

Therefore, producer's accuracy makes a statement about the underestimation (completeness) of a classification. In contrast, User's accuracy (U_i) specifies the overestimation (reliability) of the respective class as it can be expressed as the ratio between correctly classified reference points (P_{ii}) and total map class labels (P_{i+}) (Congalton, 1991):

(3) $$U_i = p_{ii}/p_{i+}$$

Overall Accuracy (OA) is the sum of correctly classified reference points over all classes (q) divided by the total number of reference points (P_{jj}):

(4) $$OA = \sum_{i=1}^{q} p_{jj}$$

These accuracy metrics can only provide a rough estimation, since accuracy variabilities and class proportions of the assessed map are ignored. Therefore, accuracy measures must be adapted to the sampling design. In "standard random stratified sampling" stratified estimators are applied to account for different area proportions of classes ($\hat{p}_{ij}$) within the map. It can be calculated as

(5) $$\hat{p}_{ij} = W_i \frac{n_{ij}}{n_{i+}}$$

where W_i is the proportion of area mapped as class i, n_{ij} is the number of samples in a cell (i,j) of the error matrix and n_{i+} is the number of samples used to calculate the respective parameter (Stehman and Foody, 2019). Using class proportions to evaluate a map facilitates a weighted assessment of the map's accuracies. By substituting p_{ij} with $\hat{p}_{ij}$ in formulas (2)-(4) and accuracy measures are recalculated (Card, 1982). Consequently, all accuracy measures of the analysis now reflect class proportions within the map. The estimator for overall accuracy then is:

(6) $\hat{O} = \sum_{i=1}^{q} \hat{p}_{jj} = \sum_{i=1}^{q} W_i \frac{n_{ij}}{n_{i+}}$

producer's accuracy is:

(7) $\widehat{P_j} = \hat{p}_{jj} / \hat{p}_{+j}$

and user's accuracy is:

(8) $\widehat{U_\iota} = \hat{p}_{ii} / \hat{p}_{i+}$

3 Results

This chapter describes classification results for the presented approach. First, training performance of the UNet model is presented for two consecutive training iterations. The second model is afterwards used to produce a LULC classification for the entire study area ("Western European broadleaf forests"). For the resulting map, an accuracy assessment is performed to obtain classification performance. Finally, multiple classifications are described in detail, where ground truth data was available. Consequentially, findings of this chapter provide a basis for answering the research questions mentioned in the beginning.

3.1 Training Performance

Evaluating the performance of the model during and after training indicates to which extent it's learning to classify. Accuracy measures describe the agreement between predicted classes and those used for training. Since the reliability of the training data used is compromised by gaps and data quality (Chapter 2.1.1), accuracy measures provided by the model can only present a rough estimation of the true classification performance. After the first training, which took 17 hours (93 epochs), overall classification accuracy reached 64%. Accuracy measures increased significantly during the second training up to 88%, using transfer learning (Chapter 2.2.1). Training on the Amazon instance stopped after about 6 hours, which corresponded to 102 epochs. The development of sparse categorical accuracy over the course of both trainings is outlined in Figure 18 and final accuracy measures are presented in Table 5.

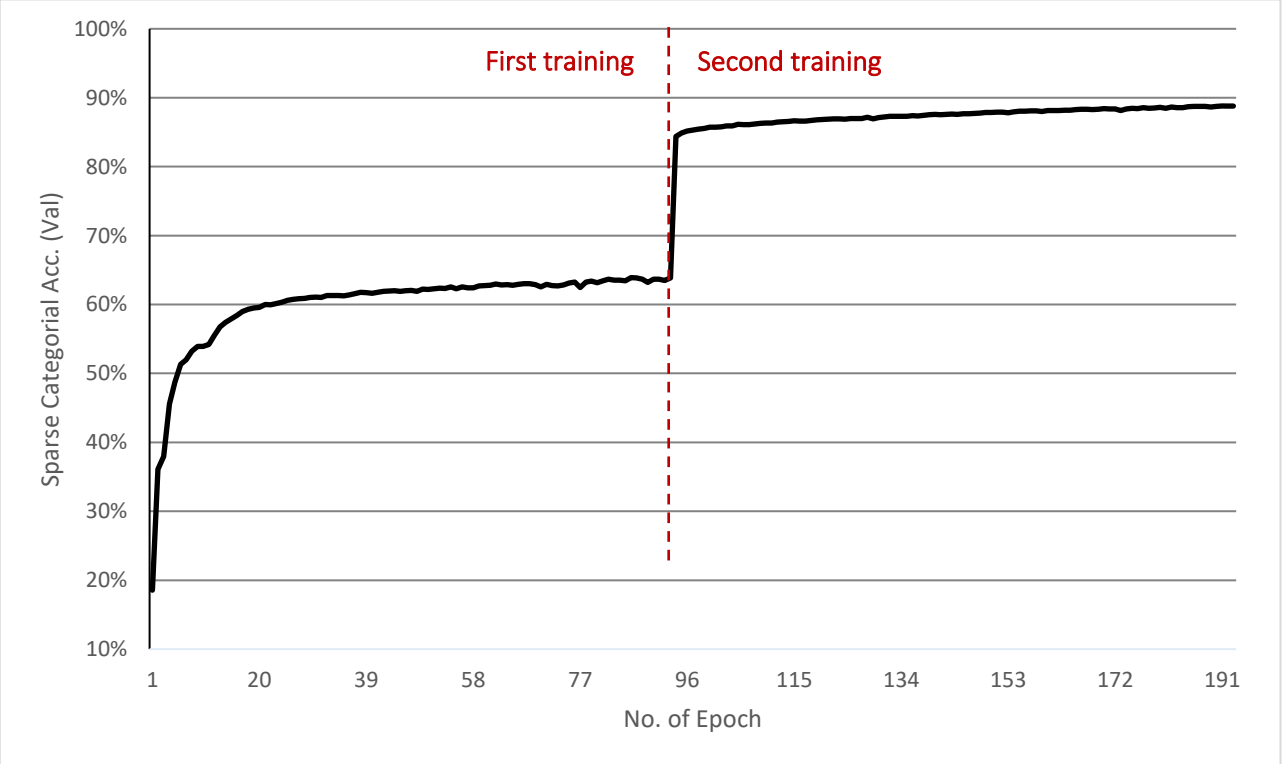

Figure 18: Progression of the Sparse Categorial Accuracy for validation dataset during training of the UNet.

Table 5: UNet final classification performances after the first and second training

Training	Loss	Sparse Categorical Accuracy (Training)	Sparse Categorical Accuracy (Validation)
1	1.41	61.6%	63.8%
2	0.43	88.1%	88.8%

3.2 Complete LULC Map

For the extent of the ecoregion "Western European broadleaf forests" a LULC classification was created by using the second and final UNet model (Chapter 3.1). First, every S2 image available within this study area for the given time period was predicted by this UNet individually. The prediction time for one training sample averaged 2.4 seconds which amounts to 6.3 hours in total. Afterwards, all predictions were merged into an extensive map depicted in Figure 19. At last, an accuracy assessment of the resulting map classification is performed to obtain its thematic accuracy. LULC classification (Figure 19) is not available across the whole study area due to characteristics of the approach. Missing classifications within the resulting map can be attributed to different steps of the task. Vertical stripes of missing data are caused by the trajectories of Sentinel-2 satellites. They mark edges of the acquisition window for complete S2 scenes. In addition, there is no classification for S2 images with a cloud cover percentage above 20% (Chapter 2.3.3). The restricted acquisition time period of the approach resulted in data loss where no suitable S2 images could be obtained. Finally, a LULC classification could not be generated, if the UNet model predicted the "no data" class for a pixel. Consequentially, only around 88.8% of the complete study area could be classified.

Figure 19 shows the complete LULC classification for the chosen study area. It also contains six small extracts (I to VI) of the classification at higher spatial resolutions. Those extracts highlight characteristics and phenomena of the classification described in the following.

Extract I

For Extract I classes Urban fabric (1.1), Pastures (2.3), Forests (3.1) and Water bodies (5) are the most dominant. Areas of the class Urban fabric (1.1) with varying sizes are distributed consistently throughout the extract. Nearly every agricultural area within it is classified as Pastures (2.3) and only a tiny fraction is considered as Arable land (2.1) or Orchards (2.2). The class Water bodies (5) often appears as small, scattered spots inside the Forests class (3.1).

Extract II

Extract II shows a high proportion of the Forest class (3.1) and some clouds interspersed with classes Urban fabric (1.1), Arable land (2.1), Pastures (2.3) and Water Bodies (5). Occasionally, class Shrub and/or herbaceous vegetation associations (3.2) is predicted by the UNet classifier inside areas classifier as class Pastures (2.3). Moreover, linear structures with deviating classifications can be identified in some spots on the left side of the extract.

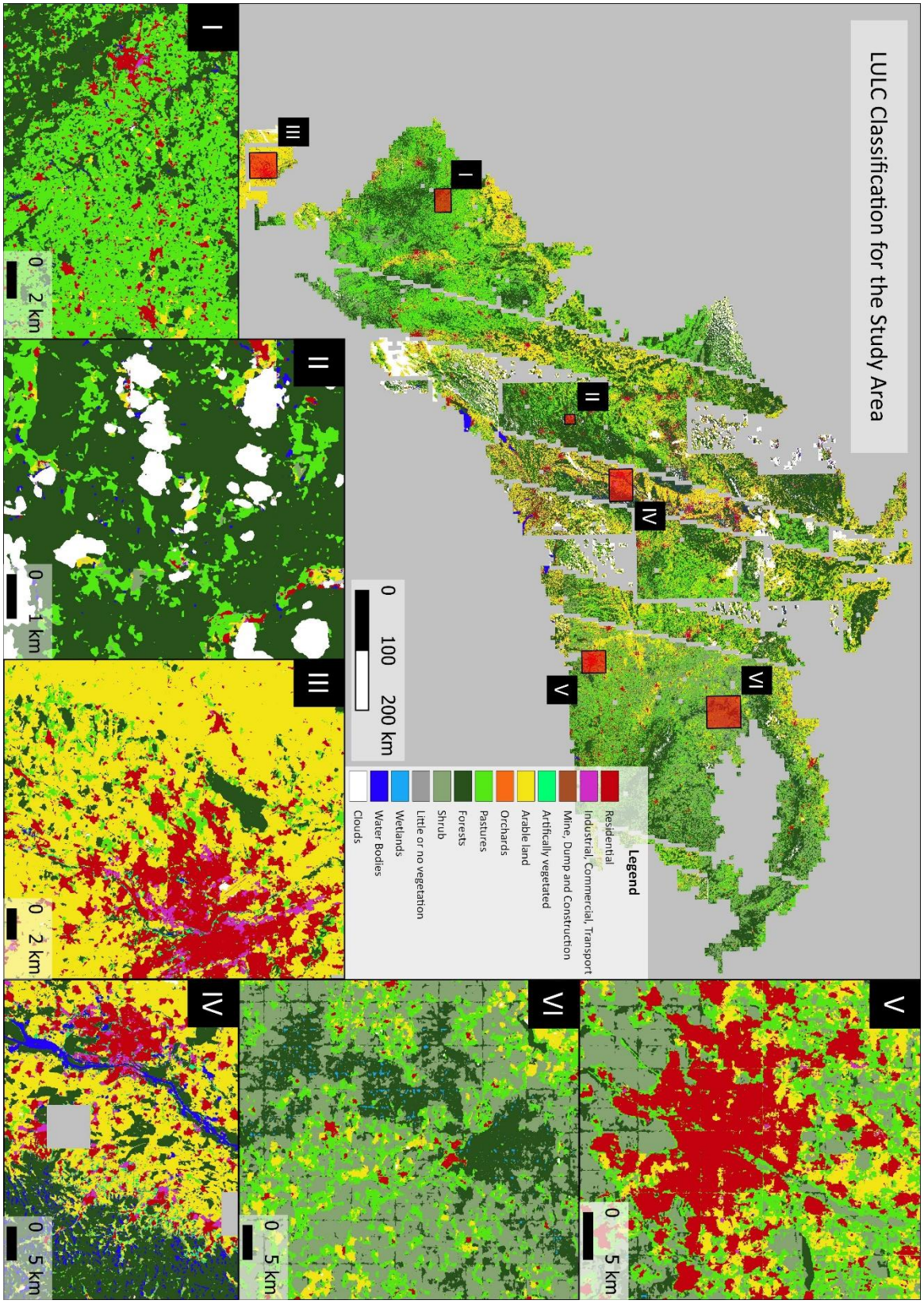

Figure 19: LULC classification for the ecoregion "Western European broadleaf forests' plus six extracts at higher spatial resolutions.

Extract III

This extract shows a large urban area with classes 1.1, 1.2 and 1.4. It is mostly surrounded by agricultural areas, which are mainly classified as Arable land (2.1). Classes 2.3, 3.1, 3.2, 5 and clouds are also visible in the extract to a lesser extent. Minor classes like 3.2 and 5 often appear as small, scattered and fragmented.

Extract IV

Extract IV covers a scene of divers landuse and landcover. On the left side an urban area characterized by classes 1.1 and 1.2 near a river-like structure of class 5 can be identified. The central part of the extract consists of arable land (2.1) interspersed with patches of forests (3.1) and urban areas (1.1 and 1.2). Rectangular shapes of no classification visible in the extract match the form of S2 images used. The right side of the scene contains a large forest area, consistently interrupted by unusual patterns of classes 3.2 and 5.

Extract V

Extract 5 shows a large agglomeration, mostly classified as Urban fabric (1.1). In contrast to urban areas in Extracts III and IV there is little to no occurrence of other urban classes (1.2, 1.3 and 1.4). Furthermore, class 3.2 takes up a lot of space here, compared to the previous extracts I to IV. Class 2.3 can often be found near class 1.1, sometimes forming a fringe around it. The extract also shows linear classification borders, which are particularly noticeable.

Extract VI

An apparent feature of this extract is its chequered classification pattern. This pattern mostly occurs on top of areas classified as Shrub and/or herbaceous vegetation associations (3.2) and happens between classes 3.1 and 3.2. In addition, agricultural and urban classes are absent for the most part of the extract, which is dominated by classes 3.1 and 3.2. Inside forest areas (3.1), the class Inland wetlands (4.1) appears as fragments, similar to class 5 in Extract IV.

A total of 942 reference points is used to assess the thematic accuracy of the map. Quantity and distribution of these is calculated using formula 1 with a confidence level of 95% (Chapter 2.3.6.1). To facilitate any calculation of accuracy metrics a minimum of 20 reference points per class is set. The number of reference points per class and the absolute and relative distribution of classes across the map is specified in Table 6. Most frequent classes in the map are Forests (3.1), Pastures (2.3) and Arable land (2.1). Several classes occupy less than 0.1% of the map respectively. These include Mine, dump and construction sites (1.3), Permanent crops and orchards (2.2) and Inland wetlands (4.1). Class Open spaces with little or no vegetation (3.3) is absent in the map. Approximately 4% of the map is predicted to be clouds and around 11% of the area is not classified at all (Table 6).

Table 6: Distribution of classes, number of pixels per class, class proportions and reference points for the LULC classification of the complete study area

CLC Class	CLC Class Name	Number of Pixels	Class Proportion	Reference Points
-	No Data	508.264.472	11.17%	-
1.1	Urban fabric	201.327.350	4.43%	51
1.2	Industrial, commercial and transport units	17.850.358	0.39%	20
1.3	Mine, dump and construction sites	1.143.113	0.03%	33
1.4	Artificial non-agricultural vegetated areas	13.505.774	0.3%	20
2.1	Arable land	806.120.276	17.72%	172
2.2	Permanent crops and orchards	47.563	0%	20
2.3	Pastures	1.082.141.359	23.79%	212
3.1	Forests	1.341.590.407	29.49%	239
3.2	Shrub and/or herbaceous vegetation associations	361.930.022	7.96%	88
3.3	Open spaces with little or no vegetation	0	0%	0
4.1	Inland wetlands	1.624.015	0.04%	20
5	Water bodies	28.097.902	0.62%	20
-	Clouds	361.930.022	4.09%	47
SUM		**4.549.587.957**	**100%**	**942**

Table 7: Confusion matrix of the accuracy assessment from the LULC map of the complete study area. Map classification is set against reference classification at the reference point location. The agreement between both classifications is presented in percent.

Classes	**1.1**	**1.2**	**1.3**	**1.4**	**2.1**	**2.2**	**2.3**	**3.1**	**3.2**	**4.1**	**5**	**clouds**
1.1	56.9	5	9.1	15	1.8	15	2.4	0.4	0	0	10	0
1.2	17.6	80	12.1	10	0.6	5	2.4	0	1.1	0	0	2.1
1.3	2	0	6.1	0	0	0	0	0	0	0	0	2.1
1.4	9.8	10	15.2	25	1.2	0	0.5	0.4	0	0	0	0
2.1	9.8	0	15.2	20	72.9	30	34.1	7.9	20.5	0	0	2.1
2.2	0	0	0	0	1.2	10	1.9	0	1.1	0	0	0
2.3	2	0	12.1	0	14.1	20	47.9	1.7	22.7	0	5	6.4
3.1	0	0	15.2	25	7.6	20	9	83.3	46.6	100	55	0
3.2	2	5	12.1	0	0.6	0	0.5	4.6	8	0	0	0
4.1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	3	5	0	0	1.4	1.7	0	0	30	0
clouds	0	0	0	0	0	0	0	0	0	0	0	87.2

Reference class labels were collected following the response design (Chapter 2.3.6.2) and are set against the map classification using an error matrix (

Table 7). Unweighted overall accuracy reaches 57.3% between map and reference classification. Best classification performance with over 80% agreement is estimated for classes Industrial, commercial and transport units (1.2), Forests (3.1) and clouds. On the other hand, multiple classes show accuracy values below 30%. These are the classes Mine, dump and construction sites (1.3), Artificial non-agricultural vegetated areas (1.4), Permanent crops and orchards (2.2), Shrub and/or herbaceous vegetation associations (3.2) and Inland wetlands (4.1). Except for classes 1.2 and 3.2, every class with a proportion of the map smaller than 1% reaches accuracy values lower than 50% at the same time (Table 6). Accordingly, most classes with a larger map proportions also show higher classification accuracies. Strong over- or underestimations towards specific classes can be observed for classes 1.1, 2.1, 2.3, 3.1, 4.1 and 5. The matrix reveals that 17.6% of the class Urban fabric (1.1) should be classified as Industrial, commercial and transport units (1.2). An indication of this overestimation can be seen in Figure 19 (Extract V). Table 7 also indicates mutual confusion between classes 2.1 and 2.3. Class 2.1 overestimates 14.1% towards class 2.3, while in turn, 34.1% of class 2.3 should be classified as class 2.1. For class 3.1 an overestimation of 7.9% relates to Arable land (2.1). Reference points collected for class 4.1 indicate a misclassification of 100% for that class. The entire class should be classified as Forests (3.1), which coincides with observations in Figure 19 (Extract VI). Differentiating Forests (3.1) from Water bodies (5) creates difficulties for the UNet classifier. Figure 19 (Extract II and IV) and the confusion matrix present this confusion both visually and in numbers. Despite its comparatively high proportion of the map (Table 6), Shrub and/or herbaceous vegetation associations (3.2) shows very low classification accuracy at 8 %. Overestimation of this class happens towards multiple classes, primarily 2.1, 2.3 and 3.1. This result is consistent with observations of Figure 19 (Extract VI).

With the chosen sampling design any accuracy measures derived from the confusion matrix must be adapted to account for different class proportions within the map. Estimated producer's and user's accuracies plus confidence intervals are illustrated class-wise in Figure 20.

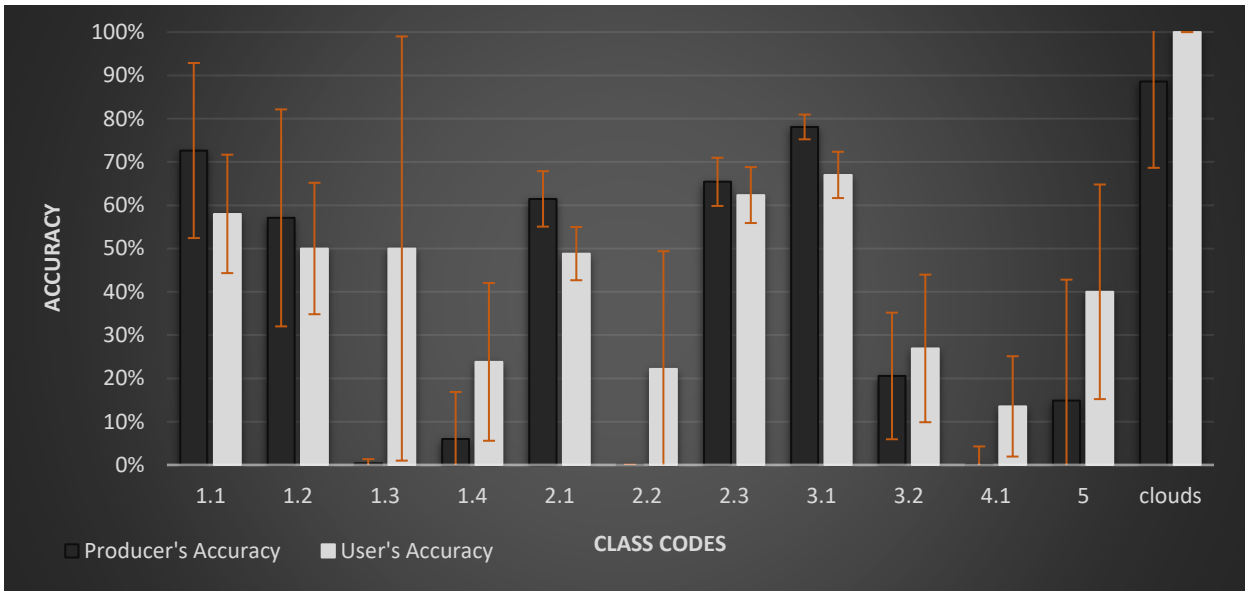

Figure 20: Corrected producer's and user's accuracy values plus confidence intervals for each class derived from the accuracy assessment of the LULC map covering the complete study area.

After recalculating accuracy measures, weighted overall accuracy reaches 62.2%. In Figure 20 producer's accuracies above 70% are found for classes Urban fabric (1.1), Forests (3.1) and clouds. This stands in stark contrast to low producer's accuracies of classes Mine, dump and construction sites (1.3), Artificial non-agricultural vegetated areas (1.4), Permanent crops and orchards (2.2), Shrub and/or herbaceous vegetation associations (3.2), Inland Wetlands (4.1) and Water bodies (5), which do not exceed 20%. Highest user's accuracy is estimated for classes Pastures (2.3), Forests (3.1) and clouds. The cloud class stands out, since it shows perfect reliability having 100% user's accuracy. However, user's accuracy values don't exceed 67% for any other class. Inland wetland (4.1) is the only class with user's accuracy below 20%. In general, producer's accuracies can peak higher than user's accuracies, but also drop lower for some classes. In turn, user's accuracies behave more stable overall, not varying as much. Using corrected measures impacts the accuracy assessment in many ways. This is expressed by differences between confusion matrix (Table 7) and corrected producer's accuracies (Figure 20). Here, estimated producer's accuracy values increases by more than 10% for classes 1.1, 2.3 and 3.2. On the contrary, producer's accuracy values decreases by more than 10% for classes 1.2, 1.4 and 5. In addition, a connection between confidence intervals and class proportions (Table 6) can be established. The three largest classes, 2.1, 2.3 and 3.1, all show confidence intervals smaller than 7%. On the contrary, classes covering less than 1% of the map's area have larger confidence intervals of more than 10% up to 48% (class 1.3). Just as for the confusion matrix, accuracy values are lower for classes with small class proportions.

3.3 Reference LULC Map

In this chapter multiple classifications for the same area are extensively compared to the reference dataset (Ground Truth) to work out strengths and weaknesses of each classification. Although findings and phenomena are only partly applicable to the complete study area, they can still provide valuable hints towards shortcomings and challenges of the respective classification. An illustration of applied classifications and reference dataset is presented in Figure 21.

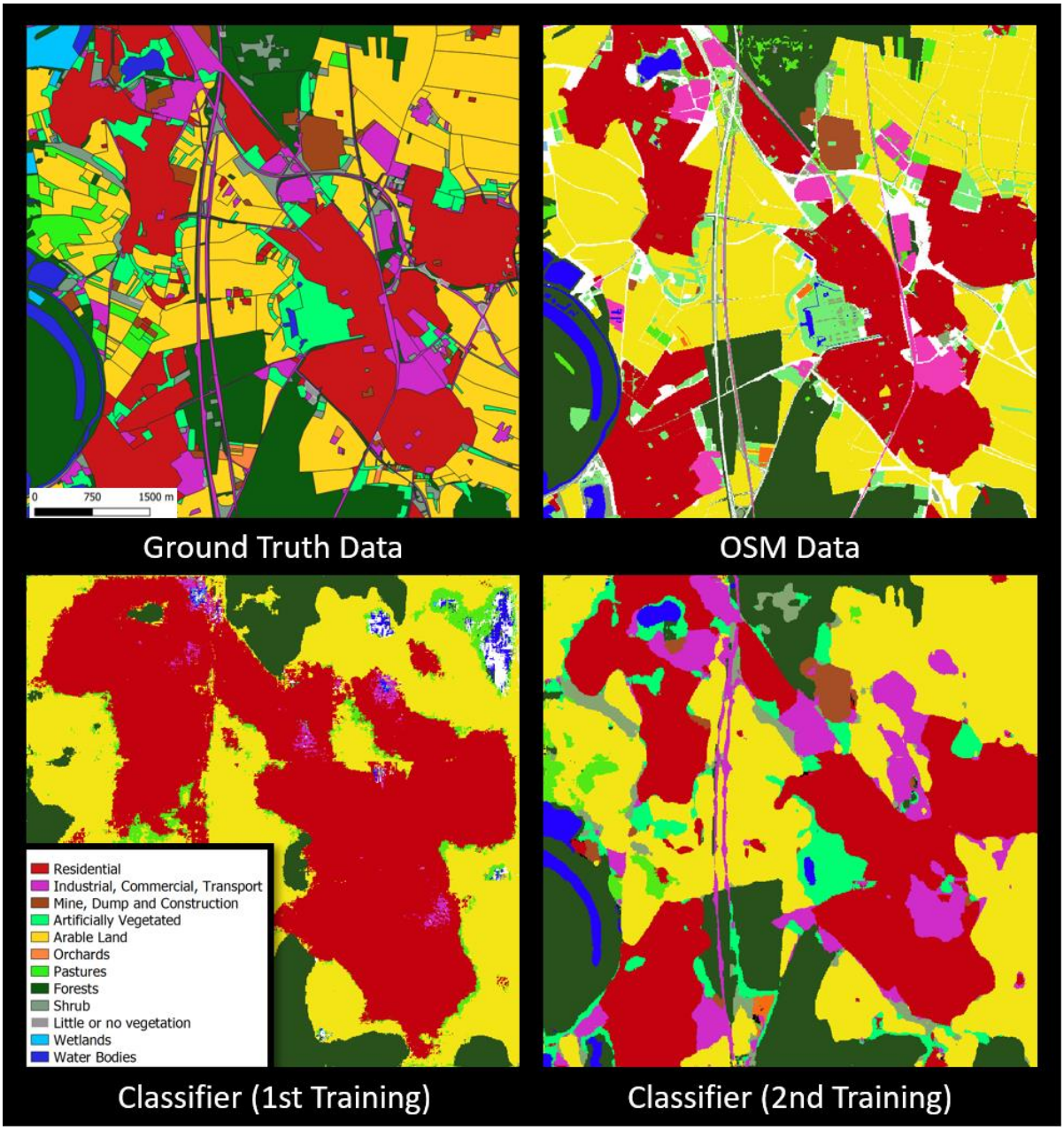

Figure 21: Visual comparison between different classifications of the Reference Dataset (Upper left). Upper right: OSM data classified using legend harmonization. Lower left: Classification predicted by the UNet after first training on the complete training dataset. Lower right: Predicted classification after training the UNet classifier on a selected subset of the training dataset (second training).

The figure suggests a strong similarity between ground truth (upper left) and underlying OSM data, whereby sporadic data gaps are visible for the OSM-based classification (upper right). The prediction of the UNet classifier from the first training (lower left) shows a low level of detail and lacks class variance. In addition, this classification is interspersed with artefacts and inaccuracies at class boundaries. In contrast to that, the UNet classifier after the second training (lower right) provides a higher level of detail and class variance. The granularity of the classification has improved visibly, whereas classification impurities decreased within the reference area.

Figure 22 highlights differences between ground truth data (reference dataset) and two selected classifications. The first one is the classification derived from OSM data using legend harmonization (Chapter 2.3.3), the other one is predicted by the UNet after the second training (Chapter 3.1). Grey areas show an agreement between classifications, whereby other colours reveal the respective class assignment for areas of disagreement.

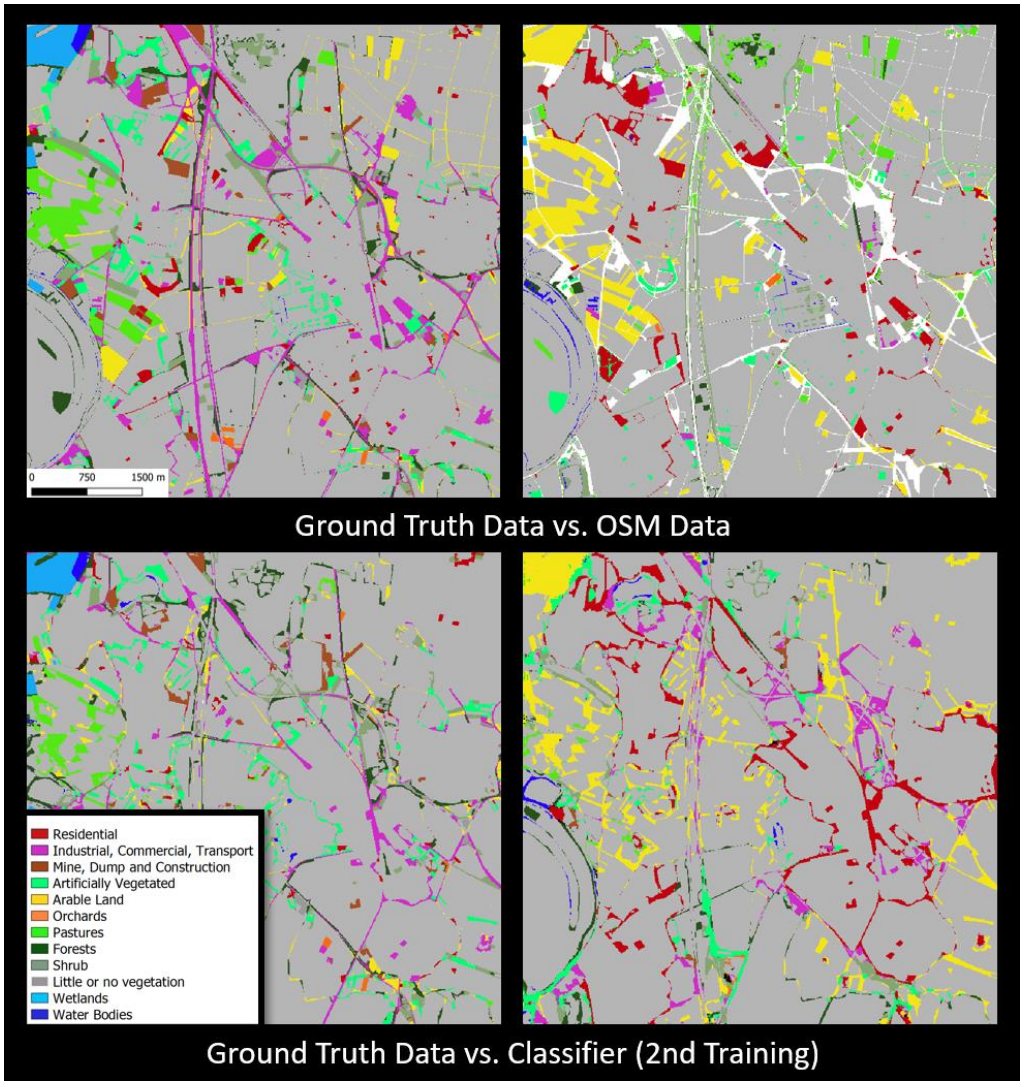

Figure 22: Reference dataset vs OSM data and vs. classification of the UNet model after the second training. Coloured areas show differences through the class assignment in the respective dataset, whereas grey colour indicates areas of agreement.

Ground truth and OSM classification visibly disagree for agricultural and infrastructural areas (Figure 22). A large part of the data gaps in the OSM classification (upper right) appears to correspond with the class Industrial, commercial and transport units (1.2) in the reference dataset (upper left). The prediction of the classifier (lower right) in comparison to the reference dataset (lower left) shows the strongest discrepancy at the borders of the UNet classification, resulting in outline effects at the edges of classes. To be able to quantify the level and direction of disagreement and between classifications with great detail, two confusion matrixes are employed.

Table 8: Confusion matrix of ground truth (reference dataset) vs. OSM classification showing class assignments in percent. Numbers in yellow boxes show the proportion of agreement for the respective class. A similar confusion matrix with absolute pixel numbers is available in the Appendix of this work.

Classes	0	1.1	1.2	1.3	1.4	2.1	2.2	2.3	3.1	3.2	3.3	4.1	5
0	0.2	0	0	0	0	0	0	0.1	0	0.1	0	0	0
1.1	5.9	90	0.2	0.9	7.6	0.8	0	0.9	0.2	3.2	0	0	0.3
1.2	36.1	3.6	90.9	0.1	2.8	1.2	0	9.6	0.5	23.8	0	0	1.5
1.3	2.5	1.2	4.3	97.5	1	0.4	0.4	6.6	0.2	0.6	18.3	0	0
1.4	13.7	2.4	0.6	0.2	76.3	1.1	0	12.3	0.6	15.7	20.4	0	3.5
2.1	14.6	1	0.2	0.1	3.2	84.1	44.6	23.3	0.3	7	0	0	0
2.2	0.2	0	0	0	1.3	0.3	54	0	0	0.6	0	0	0
2.3	1.9	0	0	0	0.1	5.9	0.9	8.5	0	0.4	0	0	0
3.1	9.3	0.8	2.3	1.1	4.4	0.6	0	14.4	95.5	31.4	3.2	100	7.8
3.2	12.3	0.9	1.4	0	1.9	2.7	0	21.8	2	15.9	58.1	0	0.6
3.3	1.4	0.1	0.1	0	1.1	0.2	0	2.5	0.1	0.4	0	0	0
4.1	2	0	0	0	0	2.3	0	0	0.3	0.1	0	0	0.9
5	0	0	0	0	0.3	0.4	0	0	0.4	0.8	0	0	85.4

Comparing ground truth and OSM classifications (Table 8) an agreement of 77.4% between the two classifications can be observed. The OSM classification shows a no data proportion of 7.9% for the whole area (see Appendix), while its largest proportion (36.1%) is assigned to Industrial, commercial and transport units (1.2) in the reference dataset. Overall best performing classes with over 90% agreement are Urban fabric (1.1), Industrial, commercial and transport units (1.2), Mine, dump and construction sites (1.3) and Forests (3.1). Accuracy is lower than 15% for classes Pastures (2.3), Shrub and/or herbaceous vegetation associations (3.2), Open spaces with little or no vegetation (3.3) and Inland wetlands (4.1). However, when considering class proportions, less than 1% of the area is classified as Open spaces with little or no vegetation (3.3) or Inland wetlands (4.1) in the reference dataset (see Appendix). The OSM classification shows the highest absolute disagreement for class Arable land (2.1). Here, an overestimation of 5.8% relates to the class Pastures (2.3) and is clearly visible in the left side of Figure 22. Class Urban fabric (1.1) is also overestimated in the OSM classification and relates to class 1.2 (3.6%) and 1.4 (2.4%). Other areas of disagreement often appear as fragments at the very edges of continuous classifications like rivers, residential areas and streets. Misclassification is common in the context of small-scale structures like standalone buildings inside large continuous areas (Figure 22). If gaps present in the OSM classification are disregarded for the calculation of accuracy measures, OA reaches 83% inside the reference area.

Table 9: Confusion matrix of ground truth (reference dataset) vs. predicted classification (UNet trained on subset) showing class assignments in percent. Numbers in yellow boxes show the proportion of agreement per class. A similar confusion matrix with absolute pixel numbers is available in the Appendix of this work.

Class	no data	1.1	1.2	1.3	1.4	2.1	2.2	2.3	3.1	3.2	3.3	4.1	5
no data	0.6	0	0.1	0	0	0	0	0	0	0	0	0	0
1.1	0.3	86.8	1.8	0.6	3.2	0.9	0	0.2	0	0.8	0	0	0.2
1.2	7.6	4.7	70.5	4.2	5.9	2.1	4.3	0.2	1.2	8	0	0	0.1
1.3	12.3	0.8	4.3	80.8	0.4	0.6	3.6	3.2	0	1.3	0	0	0
1.4	6.6	3.7	3.2	3.4	66.3	2.8	0	1.1	0.7	7.5	0	0	3.5
2.1	28.8	0.8	4.9	1.3	3.3	83.4	18	8.1	0.8	13.4	0	0	0
2.2	6.3	0	0.3	0	0.4	0.2	71.9	0	0	1.8	0	0	0
2.3	0.9	0	0	0	0.3	3.3	0	78.8	0.1	4.1	0	0	0
3.1	13	1.5	5.4	5.9	9.6	1.7	0	6.7	93.7	14	0	0	10.1
3.2	16.8	1.3	7.3	3.8	6.7	2	1.7	0	0.9	48.7	0	0	0
3.3	0	0.2	2.3	0	0.1	0.2	0	0	0.1	0.2	0	0	0
4.1	0	0	0	0	0.5	2.5	0	1.5	0.4	0.2	0	0	0
5	6.6	0	0	0	3.3	0.3	0.6	0.2	2.1	0	0	0	86.2

The UNet model can classify the reference area with an OA of 82.9%. Only a very small number of pixels is predicted as no data (316 pixels, see Appendix). Best performance is reached for classes Urban fabric (1.1) Forests (3.1) and Water bodies (5), while other classes like Artificial non-agricultural vegetated areas (1.4) and Shrub and/or herbaceous vegetation associations (3.2) have lower agreements (Table 9). Just like the classification derived from OSM data (Table 8) classes 3.3 and 4.1 are omitted in this classification. In contrast to that, agreement increased for many classes compared to the OSM data classification. Class Pastures (2.3) shows the biggest improvement with an accuracy of 78.8%, coming from 8.5% for the OSM classification. Also, accuracy for class Shrub and/or herbaceous vegetation associations (3.2) increased from 15.9% to 48.7% between the two classifications. Figure 22 visualizes over- and underestimation of classes specified in Table 9.

Although performance of class Urban fabric (1.1) is high with 86.8%, an apparent overestimation of this class by the model can be observed (Figure 22 – bottom right). More than 4% of the pixels would belong to class Industrial, commercial and transport units (1.2) and another 3.7% of its pixels should be classified as Artificial non-agricultural vegetated areas (1.4). When looking at the spatial distribution, overestimation mainly happens at edges and inside the Urban fabric (1.1) area. Especially small-scale structures like streets and commercial areas inside larger residential areas are not considered by the model. Neighbouring Artificial non-agricultural vegetated areas (1.4) like gardens and parks also cannot be distinguished clearly from the Urban fabric (1.1) class, which further contributes to the overestimation. A second striking overestimation by the model relates to class Arable land (2.1), however, unlike in the OSM classification, overestimation is more distributed across multiple classes for the UNet classification. Finally, Artificial non-agricultural vegetated areas (1.4) are not only underestimated, but also overestimated for some areas, which makes it a particularly volatile class. Overestimation for this class happens especially towards classes Forests (3.1) with 9.6%, Shrub and/or herbaceous vegetation associations (3.2) with 6.7% and Industrial, commercial and transport units (1.2) with 5.9%.

4 Discussion

The previous chapter displayed that the proposed method produced a weighted overall accuracy of 62.2%. Accuracy values vary largely among classes. Some LULC classes, like Forests (3.1), Urban fabric (1.1) and clouds can be derived from S2 data and OSM features with satisfactory accuracy (> 70%), while others cannot be recognized with comparable performance (Figure 20, Table 7). The results show multiple explicit class confusions that could partly originate from their similarity in terms of spectral signatures. Strong intensity of class confusion occurred between classes Arable land (2.1) and Pastures (2.3) (Table 7). Both are large classes linked to anthropogenic cultivation practice and are therefore more intensely effected by seasonality than other LULC classes (Kussul et al., 2017). Merging those two classes into one yield higher overall accuracy as confusion is cut. When considering class proportions to (re-)calculate accuracy measures, larger classes tend to profit from it by increasing in terms of accuracy, whereas smaller class tend to drop in accuracy (Figure 20, Table 7). A weighted accuracy assessment was especially important for this approach, because of imbalanced class proportions within the study area (Table 6). The size of the study area (492.329km²) and its abundance of OSM data (92% coverage, see appendix) could be one reason why there is a 20.7% gap between study and reference area when it comes to overall accuracy values. In a larger classification area heterogeneity of classes typically increases, which can blur class specific feature space (Mellor et al., 2015). This approach attempts to deal with this phenomenon by restricting the study area to an ecoregion (Chapter 2.3.1). Previous studies also revealed that class imbalances during the training of a ML classifier can deteriorate classification accuracy significantly (Mellor et al., 2015; Thanh Noi and Kappas, 2018). However, the impact of unbalanced training datasets has not been investigated for LULC applications in connections with FCN models yet. Resulting accuracy metrics and their distribution among LULC class strongly corresponds to comparable studies in this field (Fonte et al., 2017; Schultz et al., 2017). Those studies achieved over 80% OA using a Random Forest classifier in small (< 350km²), predominantly urban areas with high OSM data density. Strong parallels to characteristics of the reference area (Chapter 3.3) suggest that the presented approach can be used competitively in a similar setting. Potential shortcomings and error sources have to be identified and addressed where possible. Therefore, the following sections focus on those at different stages the workflow.

4.1 Data Characteristics

Among other things, classification performance always depends on the quality of training data. This is especially important in the field of ML, where a lack of data quality influences both training and prediction of a classifier (Oreski et al., 2017). The three datasets used in this work (Chapter 2.1) can be affected by quality issues for several different reasons.

Applying OpenStreetMap data for LULC mapping is linked to challenges. OSM can only provide an incomplete land use and land cover estimation for a given area, due to its heterogeneity regarding all five dimensions of geospatial quality (ISO, 2013; Neis and Zielstra, 2014; Schultz et al., 2017; Zielstra and Zipf, 2010). Quality control, which systematically identifies and filters out errors in OSM data, could be used to tackle its shortcomings. In this work, quality control in relation to completeness was addressed using regulation of OSM data density, since it's very different across LULC-related OSM features (Chapter 2.3.4). Forest and residential polygons, for example, often cover large, continuous areas, whereas small-scale structures like ponds and

construction sites are often surrounded by or are near data gaps. Thus, increasing data regulation might lead to more unbalanced class proportions, because small-scale OSM features are more likely to be disregarded in the process. Therefore, finding a suitable data density threshold, adapted to the data situation within the study area, can facilitate a more accurate LULC classification. OSM data is constantly changing, so creating a LULC map with one homogenous timestamp is challenging. This inconsistent temporal resolution inevitably reduces temporal accuracy of any derived map. To minimize errors caused by time-related deviation of OSM data, the Ohsome REST API was used to extract LULC-related OSM data for any point in time (Chapter 2.3.1). Positional accuracy of OSM data was addressed by removing any occurring overlaps following (Schultz et al., 2017). However, this robust approach can omit relevant settings and still needs to be evaluated since there are other options for dealing with overlaps in OSM data (Fonte et al., 2016). In summary, quality issues related to OSM data are subject to ongoing research and difficult to completely resolve due to the nature of VGI.

Sentinel-2 images form the second dataset in this work. Like OSM data, they play a vital role in this work by training the DL classifier. Therefore, quality of S2 data is equally important to facilitate an accurate LULC classification. Although S2 data used in this approach was subject to a preprocessing conducted by the ESA, S2 images still contained variable illumination conditions, truncations and highly reflective pixels. Truncations were addressed by removing incomplete images before the training process (Chapter 2.2.3). To compensate for brightness irregularities between images, batch normalization was applied in this work (Chapter 2.3.4). However, the success of those measures remains doubtful with regard to characteristics and phenomena observable in the LULC map of the study area (Figure 19). Besides regional differences of landuse and landcover, variable illumination conditions influence the way a DL classifier learns and interprets separate LULC classes (Kussul et al., 2017). This may aggravate the differentiation of classes for the classifier, which potentially leads to areas of (mis-)classification and classification artefacts. Classes with similar spectral signatures like Forests (3.1) and Water bodies (5) are particularly prone to confusion, as can be seen in Figure 19 (Extracts I,II and IV) and Table 7. In previous studies this challenge was addressed by using time-series image data, which also resolves the problem of a discontinuous LULC map (Guo et al., 2018; Kussul et al., 2017). However, creating and interpreting maps, which are made from a mosaic of multitemporal data comes with its own challenges. A multitemporal map normally requires more S2 images from a longer time period, potentially disregarding seasonal effects of LULC (Yuan et al., 2005). It may also suffer from noise effects and longer update cycles (Leinenkugel et al., 2019; Ndikumana et al., 2018). Using Seninel-1 radar images instead of Sentinel-2 images also resolves this problem, since radar seamlessly captures the earth's surface regardless of clouds, rain and time of day. Nevertheless, Sentinel-1 images have several disadvantages compared to Sentinel-2, mostly due to the nature of radar data. These include, speckle effects, limited availability, inconsistent acquisition strategies and complex data structures (Torbick et al., 2017). Other studies explored the potential of very high resolution RS data (Volpi and Tuia, 2017; Wurm et al., 2019) for LULC classification with promising results in small-scale scenarios. Also, utilizing additional spectral bands and common RS indices for LULC classification has proven to be effective in similar scenarios and could present an improvement for this approach in the future (Leinenkugel et al., 2019; Nguyen et al., 2018; Thanh Noi and Kappas, 2018) .

The two created reference datasets may have shortcomings, which can impact findings described previously (Chapter 3). Since both datasets were gathered by the means of manual digitization, labelling mistakes cannot be excluded (Leinenkugel et al., 2019). Time gaps between background and S2 imagery can also compromise the quality of reference data, although they were kept short in this approach (Chapter 2.3.6.2). Possible inaccuracies of class definitions provided by the labelling protocol (see Appendix) may lead to classification errors as well. In addition, potential quality issues occur when it comes to number and distribution of reference points used in the accuracy assessment. Table 6 shows that multiple small classes were assessed with not more than 20 reference points. This can affect accuracy measures like reliability (user's accuracy) of a class and confidence intervals (Figure 20). Reference datasets are always estimations of the true classification and should not be considered perfect, since human errors are not completely avoidable when creating them (Stehman and Foody, 2019).

4.2 Preprocessing

Preprocessing steps implemented for this approach can be another factor impacting the results of this work (Chapter 2.3.3). Potential errors can emerge from the legend harmonization between OSM values and Corine Land Cover (CLC) classes. Research shows, that the process of translating OSM values to LULC classes is associated with ambiguities and further difficulties (Arsanjani and Vaz, 2015; Estima and Painho, 2013; Fonte et al., 2019; Schultz et al., 2017). In this context, three central questions are raised:

1. Which OSM features should be used for the creation of a LULC map?

First, it should be decided, which type of OSM features are employed. Using only polygonal features avoids additional preprocessing steps at the cost of losing potentially relevant LULC features (Arsanjani and Vaz, 2015; Schultz et al., 2017). However, to be able to include point or line features, additional steps, like the conversion of lines into polygons or the extraction of information from points inside polygons, have to be performed (Fonte et al., 2019). This effort comes with challenges, since setting up additional processing steps and parameters can potentially present new error sources. In this context, Figure 21 suggests that utilizing line features like roads and highways in this work could be beneficial, since it might increase annotation data density with respect to class Industrial, commercial and transport units (1.2). Lastly, chosen OSM features should be analysed regarding their relevance for the respective LULC application. Not every OSM feature is needed and wanted when creating a LULC map, therefore a common approach is to filter out OSM features by means of their attributes (Arsanjani and Vaz, 2015; Fonte et al., 2017, 2019, 2016; Schultz et al., 2017). The majority of remaining OSM features is similar across all mentioned studies, but there are still discrepancies due to different filtering approaches. In the end, these decisions of inclusion or exclusion of OSM features change the data basis for consecutive steps and thus also impact classification performance.

2. Which conversion rules can be established? And how?

Because OSM data was not explicitly created for the purpose of mapping landuse or landcover, a conversion from relevant OSM tags, keys or values to a designated LULC legend is needed. This translation can be based

on the description of tags in the OSM Wiki and nomenclatures of the respective LULC product (Fonte et al., 2019). Moreover, prior studies can support remaining translation decisions and can help to identify ambiguities and uncertainties (Arsanjani and Vaz, 2015; Schultz et al., 2017). However, there has been no comprehensive, systematic and empirical evaluation of individual conversion rules so far, which makes it difficult to estimate the magnitude of conversion mistakes and its potential for improvement.

3. How can overlapping OSM features be addressed?

Dealing with overlaps due to insufficient positional accuracy of OSM features is addressed in three different manners by existing studies. One way is to always preserve the smaller polygon respectively in case overlap happens (Schultz et al., 2017). Other approaches include assigning priorities to classes (Fonte et al., 2019) and using automated topological cleaning (Arsanjani and Vaz, 2015). The influence of all three methods on classification performance has not been evaluated and therefore presents another potential error source.

In this approach, other potential error sources during preprocessing of OSM data include rasterization inaccuracies and cloud detection errors. Due to rasterization results (Figure 16, Figure 19, Figure 21) and high accuracy values for the cloud class (Figure 20), it can be concluded that those effects are negligible.

4.3 Setup and Training

Finally, setup and training of the FCN classifier present the last key aspect of this work. Some of the shortcomings identified in the results section can completely or partially be explained with configuration decisions, model or workflow characteristics. FCN models generally require a specific input size across all training data to train and predict images (Henry et al., 2019; Long et al., 2015). To facilitate the use of input images with various dimensions different approaches have been proposed by previous studies. One way is to resize all images to specific dimensions (e.g. 512*512 pixels), which can be suitable if dimensions do not vary largely across the dataset (Othman et al., 2016; Penatti et al., 2015). Another way is to use smaller subsets of images, which can be called image patches (Fu et al., 2017). Extracts V and VI of Figure 19 clearly show edge effects using image patches for this work. These include misclassifications and salt and pepper effects (Figure 19). One possible explanation for this is, that each patch (256*256 pixels) is interpreted in isolation by the classifier, which has no information about adjacent patches and classes. This missing context information could play alongside class similarities and insufficient discrimination skills of a classifier to create those effects for some patches. Increasing the ability of a classifier to distinguish classes from each other could help dealing with this challenge. One approach could be to increase the patch size, which allows a DL model to consider more and increasingly complex features, which would increase computational costs as well (Fu et al., 2017). Another strategy could be to support the DL classifier in the process of differentiating classes more reliably, by improving training data or training parameters. Future research could also investigate the potential of using different, more recent DL models with regard to edge effects and classification accuracy.

Using OSM data involves dealing with a lack of completeness (Neis and Zielstra, 2014; Zielstra and Zipf, 2010) and restricting this data source to LULC-related features aggravates this issue. While Keras and Tensorflow are among the biggest DL Python libraries, their semantic segmentation models do not support ignoring data gaps

so far. Consequentially, missing data in annotation images is always interpreted an independent class during the training process. In this way, results of this work contain a small amount of no data pixels (Figure 19, Table 9), whereby most of the no data class in the complete study area (Table 6) stems from missing S2 data, which is pointed out in Figure 19. By setting a reasonable data density (Chapter 2.3.4), misclassification towards the no data class was minimized in this work. Nevertheless, resolving this technical obstacle would likely increase classification performance.

As part of the second training, improving underlying training data could lead to significantly better results. In this context, focus could be set on:

- Increasing amount and spatial distribution of samples across the study area (Dertat, 2017)
- Creating a more balanced dataset by selecting class-specific areas (Leinenkugel et al., 2019)
- Introducing quality control measures to reduce initial misconceptions.

Ideally, combining all three measures supports training the DL classifier most effectively, but takes additional effort and research work. In the process, manual work should be avoided to facilitate reproducibility, systematization and faster implementation. After the map creation, additional post-processing methods could also help to increase classification accuracy. The success of post-processing methods in connection with LULC maps is confirmed by several studies and part of current research efforts (Fu et al., 2017; Henry et al., 2019; Kussul et al., 2017). Most prominently, Conditional Random Fields (CRFs) are applied to help mitigate salt and pepper effects and refine class boundaries (Fu et al., 2017). Looking at outline effects in Figure 22 or class artefacts in Figure 19 (Extract IV and VI) CRFs or other post-processing techniques have the potential to improve the results of this work.

5 Conclusion

This thesis presented an approach for creating land use and land cover maps from open data sources using Deep Learning methods. DL techniques were applied since they incorporate higher-level features, including textures and geometric features and can therefore be considered superior to traditional RS classifiers in terms of mapping performance (Othman et al., 2016). Focus of the work was placed on generation and assessment of resulting maps with CLC legend for the ecoregion "Western European broadleaf forests". In total, 9200 Sentinel-2 scenes ranging from June to August 2018, as well as corresponding LULC-related OpenStreetMap features were acquired for that task. In addition, reference data was produced in the context of a Volunteered Geographical Information workshop by more than 40 student contributors in cooperation with the University of Jena in July 2019 to facilitate further analysis of results. The following process of creating two LULC map, including preceding acquisition and processing steps, was explained step by step for reproducibility.

Resulting maps were described and quantitatively assessed by conducting two accuracy assessments. Here, overall accuracy reached 62.2% for the study area and 82.9% for the reference area. Accuracies varied largely between 0% to 88% and 0% to 95% respectively. Best performance was reached for classes Forests (3.1) and Urban fabric (1.1) inside the study area. On the contrary, classes Mine, dump and construction sites (1.3), Artificial non-agricultural vegetated areas (1.4) and Inland wetlands (4.1) performed poorly with less than 20% accuracy. These accuracy outcomes correlate with findings from earlier studies (Fonte et al., 2019; Schultz et al., 2017) indicating high potential of this approach.

Abundance of OSM data, size, distinct features and urban properties were found to be strong factors to why classification accuracy was much better within the reference area. In both maps, strong confusion was found between agricultural areas (classes 2.1, 2.2 and 2.3). Consequentially, merging those could increase overall accuracy of resulting maps but also reduces thematic depth. Also, a connection between small class proportions and low accuracy values could be derived. Since the UNet classifier had difficulties dealing with underrepresented classes, balancing classes in training data likely improves classification performance (Leinenkugel et al., 2019; Mellor et al., 2015). This indicates that the amount of data required to successfully train small LULC classes is inversely proportional to the amount of data provided by OSM features, where small classes are often surrounded by gaps. Thus, improving quantity and quality of underlying training data can be identified as a key objective to create more accurate classifications in future works. Other potential measures include using more spectral bands or adding band indices, reworking the translation ofy OSM values into LULC classes and introducing post-processing methods. Also, multiple specialized classifiers could be employed to address classes individually.

Although this work does not reach quality levels of regional, tailor-made LULC products, it supports the fast and simple generation of LULC information for any given region, provided that enough OSM features are present. It also allows the generation of LULC maps, while considering seasonality, variable input dimensions and cloud cover. Moreover, it explores the possibilities of using a Fully Convolutional Network in combination with transfer learning, open data and open-source software to create an automated and modular workflow from end-to-end.

References

Arsanjani, J.J., Mooney, P., Zipf, A., Schauss, A., 2015. Quality assessment of the contributed land use information from OpenStreetMap versus authoritative datasets, in: OpenStreetMap in GIScience. Springer, pp. 37–58.

Arsanjani, J.J., Vaz, E., 2015. An assessment of a collaborative mapping approach for exploring land use patterns for several European metropolises. Int. J. Appl. Earth Obs. Geoinformation 35, 329–337.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2481–2495.

Card, D.H., 1982. Using Known Map Category Marginal Frequencies to Improve Estimates of Thematic Map accuracy. Photogramm. Eng. Remote Sens. 48, 431–439.

Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015. Land use classification in remote sensing images by convolutional neural networks. ArXiv Prepr. ArXiv150800092.

Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y., 2014. Deep learning-based classification of hyperspectral data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7, 2094–2107.

Congalton, R.G., 1991. A review of assessing the accuracy of classifications of remotely sensed data. Remote Sens. Environ. 37, 35–46.

Dertat, A., 2017. Applied Deep Learning - Part 4: Convolutional Neural Networks.

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., others, 2012. Sentinel-2: ESA's optical high-resolution mission for GMES operational services. Remote Sens. Environ. 120, 25–36.

Estima, J., Painho, M., 2013. Exploratory analysis of OpenStreetMap for land use classification, in: Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information. ACM, pp. 39–46.

Fisher, P., Comber, A.J., Wadsworth, R., 2005. Land use and land cover: contradiction or complement. Re-Present. GIS 85–98.

Fonte, C., Minghini, M., Patriarca, J., Antoniou, V., See, L., Skopeliti, A., 2017. Generating up-to-date and detailed land use and land cover maps using OpenStreetMap and GlobeLand30. ISPRS Int. J. Geo-Inf. 6, 125.

Fonte, C.C., Minghini, M., Antoniou, V., See, L., Patriarca, J., Brovelli, M.A., Milcinski, G., 2016. An automated methodology for converting OSM data into a land use/cover map, in: Proceedings of the 6 Th International Conference on Cartography & GIS. pp. 462–473.

Fonte, C.C., Patriarca, J.A., Minghini, M., Antoniou, V., See, L., Brovelli, M.A., 2019. Using openstreetmap to create land use and land cover maps: Development of an application, in: Geospatial Intelligence: Concepts, Methodologies, Tools, and Applications. IGI Global, pp. 1100–1123.

Foody, G.M., 2009. Sample size determination for image classification accuracy assessment and comparison. Int. J. Remote Sens. 30, 5273–5291.

Fu, G., Liu, C., Zhou, R., Sun, T., Zhang, Q., 2017. Classification for high resolution remote sensing imagery using a fully convolutional network. Remote Sens. 9, 498.

Gao, Q., Lim, S., Jia, X., 2018. Hyperspectral image classification using convolutional neural networks and multiple feature learning. Remote Sens. 10, 299.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.

Guo, Y., Jia, X., Paull, D., 2018. Mapping of Rice Varieties with Sentinel-2 Data via Deep CNN Learning in Spectral and Time Domains, in: 2018 Digital Image Computing: Techniques and Applications (DICTA). IEEE, pp. 1–7.

Henry, C.J., Storie, C.D., Palaniappan, M., Alhassan, V., Swamy, M., Aleshinloye, D., Curtis, A., Kim, D., 2019. Automated LULC map production using deep neural networks. Int. J. Remote Sens. 40, 4416–4440.

Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18, 1527–1554.

Immitzer, M., Vuolo, F., Atzberger, C., 2016. First experience with Sentinel-2 data for crop and tree species classifications in central Europe. Remote Sens. 8, 166.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ArXiv Prepr. ArXiv150203167.

ISO, I., 2013. 19157: 2013: Geographic Information—Data Quality. ISO-Stand. Swed. SIS Stand.

Jones, K., 2008. Importance of land cover and biophysical data in landscape-based environmental assessments. N. Am. Land Cover Summit Assoc. Am. Geogr. Wash. DC USA 215, 249.

Kaggle Team, 2017. Dstl Satellite Imagery Competition, 1st Place Winner's Interview: Kyle Lee. Off. Blog Kagglecom. URL http://blog.kaggle.com/2017/04/26/dstl-satellite-imagery-competition-1st-place-winners-interview-kyle-lee/

Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A., 2017. Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci. Remote Sens. Lett. 14, 778–782.

Lamba, H., 2019. Understanding Semantic Segmentation with UNET [WWW Document]. Medium. URL https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47 (accessed 9.10.19).

Lavreniuk, M., 2017. Ensemble of Convolutional Neural Networks for Crop Classification of Sentinel-1 SAR data. Presented at the The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IEEE.

Leinenkugel, P., Deck, R., Huth, J., Ottinger, M., Mack, B., 2019. The Potential of Open Geodata for Automated Large-Scale Land Use and Land Cover Classification. Remote Sens. 11, 2249.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440.

Ludwig, I., Voss, A., Krause-Traudes, M., 2011. A Comparison of the Street Networks of Navteq and OSM in Germany, in: Advancing Geoinformation Science for a Changing World. Springer, pp. 65–84.

Marmanis, D., Datcu, M., Esch, T., Stilla, U., 2016. Deep learning earth observation classification using ImageNet pretrained networks. IEEE Geosci. Remote Sens. Lett. 13, 105–109.

Mellor, A., Boukir, S., Haywood, A., Jones, S., 2015. Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin. ISPRS J. Photogramm. Remote Sens. 105, 155–168.

Morrison, J., Olson, D.M., 2005. The Natural Vegetation Map of Europe: A Regional Source for WWF's Terrestrial Ecoregions of the World Die Karte der natürlichen Vegetation Europas als regionale Grundlage für die WWF-Karte der terrestrischen Ökoregionen der Welt. Anwend. Auswert. Kt. Nat. Veg. Eur. Appl. Anal. Map Nat. Veg. Eur. 71.

Ndikumana, E., Ho Tong Minh, D., Baghdadi, N., Courault, D., Hossard, L., 2018. Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France. Remote Sens. 10, 1217.

Neis, P., Zielstra, D., 2014. Recent developments and future trends in volunteered geographic information research: The case of OpenStreetMap. Future Internet 6, 76–106.

Nguyen, M.H., Block, J., Crawl, D., Siu, V., Bhatnagar, A., Rodriguez, F., Kwan, A., Baru, N., Altintas, I., 2018. Land Cover Classification at the Wildland Urban Interface using High-Resolution Satellite Imagery and Deep Learning, in: 2018 IEEE International Conference on Big Data (Big Data). IEEE, pp. 1632–1638.

Olson, D.M., Dinerstein, E., Wikramanayake, E.D., Burgess, N.D., Powell, G.V., Underwood, E.C., D'amico, J.A., Itoua, I., Strand, H.E., Morrison, J.C., others, 2001. Terrestrial Ecoregions of the World: A New Map of Life on EarthA new global map of terrestrial ecoregions provides an innovative tool for conserving biodiversity. BioScience 51, 933–938.

Oreski, D., Oreski, S., Klicek, B., 2017. Effects of dataset characteristics on the performance of feature selection techniques. Appl. Soft Comput. 52, 109–119.

Othman, E., Bazi, Y., Alajlan, N., Alhichri, H., Melgani, F., 2016. Using convolutional features and a sparse autoencoder for land-use scene classification. Int. J. Remote Sens. 37, 2149–2167.

Patterson, J., Gibson, A., 2017. Deep learning: A practitioner's approach. O'Reilly Media, Inc.

Penatti, O.A., Nogueira, K., Dos Santos, J.A., 2015. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 44–51.

Perez, L., Wang, J., 2017. The effectiveness of data augmentation in image classification using deep learning. ArXiv Prepr. ArXiv171204621.

Poiani, T.H., dos Santos Rocha, R., Degrossi, L.C., de Albuquerque, J.P., 2016. Potential of collaborative mapping for disaster relief: A case study of OpenStreetMap in the Nepal earthquake 2015, in: 2016 49th Hawaii International Conference on System Sciences (HICSS). IEEE, pp. 188–197.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., others, 1988. Learning representations by back-propagating errors. Cogn. Model. 5, 1.

Schultz, M., Voss, J., Auer, M., Carter, S., Zipf, A., 2017. Open land cover from OpenStreetMap and remote sensing. Int. J. Appl. Earth Obs. Geoinformation 63, 206–213.

Sentinel Online - ESA [WWW Document], 2019. . Sentin. Online - ESA. URL https://sentinel.esa.int/web/sentinel/ (accessed 11.19.19).

Shah, T., 2017. About Train, Validation and Test Sets in Machine Learning [WWW Document]. Medium. URL https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7 (accessed 8.9.19).

Shibuya, N., 2017. Up-sampling with Transposed Convolution [WWW Document]. Medium. URL https://medium.com/activating-robotic-minds/up-sampling-with-transposed-convolution-9ae4f2df52d0 (accessed 9.10.19).

Sibanda, M., Mutanga, O., Rouget, M., 2015. Examining the potential of Sentinel-2 MSI spectral resolution in quantifying above ground biomass across different fertilizer treatments. ISPRS J. Photogramm. Remote Sens. 110, 55–65.

Stehman, S.V., Foody, G.M., 2019. Key issues in rigorous accuracy assessment of land cover products. Remote Sens. Environ. 231, 111199.

Strahler, A.H., Boschetti, L., Foody, G.M., Friedl, M.A., Hansen, M.C., Herold, M., Mayaux, P., Morisette, J.T., Stehman, S.V., Woodcock, C.E., 2006. Global land cover validation: Recommendations for evaluation and accuracy assessment of global land cover maps. Eur. Communities Luxemb. 51.

Sui, D., Elwood, S., Goodchild, M., 2012. Crowdsourcing geographic knowledge: volunteered geographic information (VGI) in theory and practice. Springer Science & Business Media.

Thanh Noi, P., Kappas, M., 2018. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. Sensors 18, 18.

Torbick, N., Chowdhury, D., Salas, W., Qi, J., 2017. Monitoring rice agriculture across myanmar using time series Sentinel-1 assisted by Landsat-8 and PALSAR-2. Remote Sens. 9, 119.

Volpi, M., Tuia, D., 2017. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. IEEE Trans. Geosci. Remote Sens. 55, 881–893.

Waters, C.N., Zalasiewicz, J., Summerhayes, C., Barnosky, A.D., Poirier, C., Ga\luszka, A., Cearreta, A., Edgeworth, M., Ellis, E.C., Ellis, M., others, 2016. The Anthropocene is functionally and stratigraphically distinct from the Holocene. Science 351, aad2622.

Wiki, O., 2019a. Stats — OpenStreetMap Wiki.

Wiki, O., 2019b. Map Features — OpenStreetMap Wiki.

Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

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 J. Photogramm. Remote Sens. 150, 59–69.

Yuan, F., Sawaya, K.E., Loeffelholz, B.C., Bauer, M.E., 2005. Land cover classification and change analysis of the Twin Cities (Minnesota) Metropolitan Area by multitemporal Landsat remote sensing. Remote Sens. Environ. 98, 317–328.

Zeiler, M.D., 2012. ADADELTA: an adaptive learning rate method. ArXiv Prepr. ArXiv12125701.

Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geosci. Remote Sens. Mag. 5, 8–36.

Zielstra, D., Zipf, A., 2010. A comparative study of proprietary geodata and volunteered geographic information for Germany, in: 13th AGILE International Conference on Geographic Information Science.

Zupanc, A., 2017. Improving Cloud Detection with Machine Learning [WWW Document]. Medium. URL https://medium.com/sentinel-hub/improving-cloud-detection-with-machine-learning-c09dc5d7cf13 (accessed 8.3.19).

Appendix

1. Labelling Protocol

Code	Class	Description
1.1	Urban fabric	Areas mainly occupied by dwellings and buildings used by administrative/public utilities or collectivities, including their connected areas (associated lands, approach road network, parking-lots)
1.2	Industrial, commercial and transport units	Areas mainly occupied by industrial activities of transformation and manufacturing, trade, financial activities and services, transport infrastructures for road traffic and rail networks, airport installations, river and sea port installations, including their associated lands and access infrastructures. Includes industrial livestock rearing facilities
1.3	Mine, dump and construction sites	Artificial areas mainly occupied by extractive activities, construction sites, man-made waste dump sites and their associated lands
1.4	Artificial non-agricultural vegetated areas	Areas voluntarily created for recreational use. Includes green or recreational and leisure urban parks, sport and leisure facilities
2.1	Arable land	Lands under a rotation system used for annually harvested plants and fallow lands, which are permanently or not irrigated. Includes flooded crops such as rice fields and other inundated croplands
2.2	Permanent crops	All surfaces occupied by permanent crops, not under a rotation system. Includes ligneous crops of standards cultures for fruit production such as extensive fruit orchards, olive groves, chestnut groves, walnut groves shrub orchards such as vineyards and some specific low-system orchard plantation, espaliers and climbers.
2.3	Pastures	Lands, which are permanently used (at least 5 years) for fodder production. Includes natural or sown herbaceous species, unimproved or lightly improved meadows and grazed or mechanically harvested meadows. Regular agriculture impact influences the natural development of natural herbaceous species composition
3.1	Forests	Areas occupied by forests and woodlands with a vegetation pattern composed of native or exotic coniferous and/or deciduous trees and which can be used

		for the production of timber or other forest products. The forest trees are under normal climatic conditions higher than 5 m with a canopy closure of 30 % at least. In case of young plantation, the minimum cut-off-point is 500 subjects by ha.
3.2	Shrub and/or herbaceous vegetation associations	Temperate shrubby areas with Atlantic and alpine heaths, sub Alpine bush and tall herb communities, deciduous forest re-colonisation, hedgerows, dwarf conifers. All transitional forest stages development (regenerative and degenerative: natural development of forest – bushy formations on abandoned meadows, pastures or forest clear cut and also forest after calamities of various origin). Dry thermophilous grasslands of the lowlands, hills and mountain zone. Poor Atlantic a subAtlantic mat-grasslands of acid soils; grasslands of decalcified sands; Alpine and sub Alpine grasslands. Humid grasslands and tall herb communities; lowland and mountain mesophile pastures and hay meadows.
5.0	Water bodies	low floating aquatic vegetation with species such as Nuphar spp., Nymphaea spp., Potamageton spp. and Lemna spp.; archipelago of lakes inside land areas; water surfaces used for fresh-water fish-breeding activities; fish ponds and water reservoirs temporarily without water (seasonal lack of water, maintenance, etc.

2. Confusion matrix of the accuracy assessment from the LULC map of the complete study area. Map classification is set against reference point classification on a pixel level. Numbers in yellow boxes show the number of agreeing pixels at the reference points of each class.

Classes	1.1	1.2	1.3	1.4	2.1	2.2	2.3	3.1	3.2	4.1	5	clouds
1.1	29	1	3	3	3	3	5	1	0	0	2	0
1.2	9	16	4	2	1	1	5	0	1	0	0	1
1.3	1	0	2	0	0	0	0	0	0	0	0	1
1.4	5	2	5	5	2	0	1	1	0	0	0	0
2,1	5	0	5	4	124	6	72	19	18	0	0	1
2.2	0	0	0	0	2	2	4	0	1	0	0	0
2.3	1	0	4	0	24	4	101	4	20	0	1	3
3.1	0	0	5	5	13	4	19	199	41	20	11	0
3.2	1	1	4	0	1	0	1	11	7	0	0	0
4.1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	1	0	0	3	4	0	0	6	0
clouds	0	0	0	0	0	0	0	0	0	0	0	41

3. Confusion matrix of ground truth (Reference dataset) vs. OSM data showing class assignments pixelwise. Numbers in yellow boxes show the number of agreeing pixels per class

Classes	no data	1.1	1.2	1.3	1.4	2.1	2.2	2.3	3.1	3.2	3.3	4.1	5	SUM
no data	61	24	1	0	1	14	0	11	1	15	0	0	1	**129**
1.1	1975	99513	34	38	1515	1211	0	75	125	367	0	0	25	**104878**
1.2	12099	4026	15082	5	566	1682	0	818	328	2715	0	0	139	**37460**
1.3	839	1317	716	3985	194	627	3	559	105	71	17	0	0	**8433**
1.4	4604	2694	95	10	15238	1569	0	1049	405	1798	19	0	312	**27793**
2.1	4898	1061	30	5	629	120255	300	1986	219	795	0	0	0	**130178**
2.2	80	0	0	0	257	494	363	2	6	68	0	0	0	**1270**
2.3	629	0	2	0	24	8374	6	720	18	48	0	0	0	**9821**
3.1	3104	878	375	44	879	835	0	1230	62032	3590	3	73	704	**73747**
3.2	4115	948	236	0	377	3832	0	1853	1275	1813	54	0	53	**14556**
3.3	461	66	19	0	223	332	0	214	33	45	0	0	0	**1393**
4.1	655	3	0	0	0	3303	0	0	190	9	0	0	83	**4243**
5	8	0	0	0	62	509	0	0	240	86	0	0	7694	**8599**
SUM	**33528**	**110530**	**16590**	**4087**	**19965**	**143037**	**672**	**8517**	**64977**	**11420**	**93**	**73**	**9011**	**422500**

4. Confusion matrix of ground truth (Reference dataset) vs. predicted classification (UNet trained on subset) showing class assignments pixelwise. Numbers in yellow boxes show the number of agreeing pixels per class.

Classes	no data	1.1	1.2	1.3	1.4	2.1	2.2	2.3	3.1	3.2	3.3	4.1	5	SUM
no data	2	34	32	0	9	44	0	2	3	3	0	0	0	**129**
1.1	1	101875	652	33	797	1384	0	10	30	85	0	0	11	**104878**
1.2	24	5552	25426	234	1469	3088	31	10	804	817	0	0	5	**37460**
1.3	39	996	1545	4502	90	908	26	176	18	133	0	0	0	**8433**
1.4	21	4320	1171	191	16496	4090	0	62	432	772	0	0	238	**27793**
2.1	91	917	1768	72	833	124048	129	441	508	1370	0	0	1	**130178**
2.2	20	37	91	0	92	331	516	0	0	183	0	0	0	**1270**
2.3	3	51	6	2	80	4898	0	4292	74	415	0	0	0	**9821**
3.1	41	1811	1947	326	2386	2563	0	363	62198	1427	0	0	685	**73747**
3.2	53	1485	2624	210	1655	2914	12	0	623	4980	0	0	0	**14556**
3.3	0	238	826	0	30	236	0	0	44	19	0	0	0	**1393**
4.1	0	0	0	0	123	3768	0	82	248	22	0	0	0	**4243**
5	21	36	1	1	827	428	4	9	1405	3	0	0	5864	**8599**
SUM	**316**	**117352**	**36089**	**5571**	**24887**	**148700**	**718**	**5447**	**66387**	**10229**	**0**	**0**	**6804**	**422500**

Erklärung der Urheberschaft (DE)

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum Unterschrift