Master Thesis

submitted within the UNIGIS MSc programme
Interfaculty Department of Geoinformatics - Z_GIS
University of Salzburg

Comparison of TempCNN and LightGBM for Crop Type Classification using Sentinel-2 Imagery

by

Frank Willing, BSc
105107

A thesis submitted in partial fulfilment of the requirements of
the degree of
Master of Science (Geographical Information Science & Systems) – MSc (GIS)

Advisor:

Prof. Dr. Stefan Lang

Münster, 06.12.2020

Science pledge

By my signature below, I certify that my thesis is entirely the result of my own work. I have cited all sources I have used in my thesis and I have always indicated their origin.

Münster, 06/12/2020	F.Willing
Place, Date	Signature

Preface

This thesis is the final work of my Master study in Geoinformatics at the UNIGIS Salzburg. It is a documentation of my research work, which was done between April 2019 and May 2020. It represents the results of a study for crop type classification using state-of-the-art Machine Learning and Deep Learning techniques. This thesis is written as a manuscript-based thesis as it questions assumptions in the current state-of-the-art and it is hence, of high interest to publish the results. The manuscript is written for a possible publication in the European Journal of Remote Sensing. This open access journal has a strong focus on articles related to the use of remote sensing technologies and numerous publications with agricultural background. Therefore, it states a suitable framework for this research.

This thesis contains two main sections. The manuscript (section 1) considers all specifications of the European Journal of Remote Sensing and is ready for the submission to be published. The report in section 2 outlines all further technical and scientific developments and analysis done within this research work, that could not be included in more detail in the manuscript.

List of Contents

Science pledge I
Preface II
List of Contents III
List of Figures V
List of Tables VI
List of Code Subsets VI
List of Abbreviations VII
1. Manuscript 1
1.1. Introduction 2
1.1.1 Satellite Image Time Series Analysis 2
1.1.2 Aim and Objectives 4
1.2. Material and Methods 6
1.2.1 Study Area 6
1.2.2 Data 8
1.2.2.1 Sentinel-2 Imagery 8
1.2.2.2 Ground Reference Data 9
1.2.2.3 Dataset Partition 9
1.2.3 Classification 9
1.2.3.1 Benchmark ML Classifier 9
1.2.3.2 DL Classifier 10
1.2.3.3 Evaluation 11
1.3. Results 11
1.3.1 Crop Grouping 11
1.3.2 Model Stability on heterogenous Groups 13
1.3.3 The Impact of Feature Engineering – Indices 13
1.3.4 The Impact of Feature Engineering – temporal Resolution 14
1.3.5 Model Stability on spatial Variation 15
1.4. Discussion 17
1.5. Conclusion 19
Acknowledgment 21
References 22

2. Report 24
2.1 Software 24
2.1.1 Python 24
2.1.2 eo-learn Library 25
2.2 Data Pre-Processing 28
2.2.1 AOI 28
2.2.2 EO-Data 28
2.2.3 Reference Data 32
2.2.4 Feature Engineering 33
2.2.5 Interpolation 34
2.2.6 Sampling 35
2.3 Classification 36
2.3.1 Reference Classification 38
2.3.2 TempCNN Classification 42
2.4 Accuracy Assessment 47
References 50
Appendix 52

List of Figures

Figure 1: Study area split into train/ test units as well as external single units for spatial generalization checks and examples of LPIS and Sentinel-2 raster data (Quelle: Eigene Darstellung) 6

Figure 2: Crop distribution in alpine and foreland Austrian regions (Quelle: Eigene Darstellung) 7

Figure 3: Crop distribution for the 3 regions of the AOI (Quelle: Eigene Darstellung) 8

Figure 4: Stepwise accuracy check for single crop grouping steps related to individual crop groups (Quelle: Eigene Darstellung) 12

Figure 5: LightGBM and TempCNN performances on heterogenous groups (Quelle: Eigene Darstellung) 13

Figure 6: LightGBM and TempCNN performances on datasets including and missing spectral indices (Quelle: Eigene Darstellung) 14

Figure 7: LightGBM and TempCNN performances on 4- and 8-day resampled datasets (Quelle: Eigene Darstellung) 15

Figure 8: LightGBM and TempCNN performance on test data from outside the training area (Quelle: Eigene Darstellung) 15

Figure 9: Distribution of LPIS crop class Potatoes and respective TempCNN performance in contrast to LightGBM (Quelle: Eigene Darstellung) 16

Figure 10: Distribution of LPIS crop class Maize and respective TempCNN performance in contrast to LightGBM (Quelle: Eigene Darstellung) 17

Figure 11: Timeframes per EOPatch (Quelle: Eigene Darstellung) 31

Figure 12: LPIS vector data (Quelle: Eigene Darstellung) 32

Figure 13: Classes distribution in pixel per class for training and test dataset (Quelle: Eigene Darstellung) 36

Figure 14: Overview AI (Quelle: Eigene Darstellung) 38

Figure 15: Decision Tree (Quelle: Eigene Darstellung)..... 39

Figure 16: Level-wise growth (Quelle: Eigene Darstellung) 41

Figure 17: Leaf-wise growth (Quelle: Eigene Darstellung)..... 41

Figure 18: Artificial Neural Network (Quelle: Pelletier et al. 2019) 42

Figure 19: Temporal Convolutional Neural Network architecture (Quelle: Pelletier et al. 2019)..... 44

Figure 20: Convolution result of Gradient Extraction Filter [-1 -1 0 1 1] (Quelle: Pelletier et al. 2019)..... 45

Figure 21: Convolutional filter guidance (Quelle: Pelletier et al. 2019)..... 46

List of Tables

Table 1: EOPatch data formats..... 26

Table 2: eo-learn subpackages 27

Table 3: SCL-Classes 30

Table 4: Cloud Masking..... 31

Table 5: Crop grouping 52

List of Code Subsets

CodeSubset 1: LightGBM – Setup and training..... 41

CodeSubset 2: TempCNN – Setup and training..... 47

List of Abbreviations

AOI	Area of interest
ANN	Artificial Neural Network
CLM	Cloud Mask
CNN	Convolutional Neural Network
DL	Deep Learning
DT	Decision Tree
EO	Earth Observation
GBDT	Gradient Boosting Decision Tree
k-NN	k-Nearest Neighbour
LightGBM	Light Gradient Boosting Machine
LPIS	Land Parcel Identification System
LSTM	Long Short-term Memory
LULC	Land Use / Land Cover
MF	Marchfeld
ML	Machine Learning
MSI	Multispectral Instrument
NDWI	Normalized Difference Water Index
NDVI	Normalized Difference Vegetation Index
NIR	Near Infrared
NORM	Normalized Euclidean Distance
OÖ	Oberösterreich
ReLU	Rectified Linear Units
RF	Random Forest
RNN	Recurrent Neural Network
SCL	Scene Classification
SITS	Satellite Image Time Series
SM	Steiermark
SVM	Support Vector Machine
TempCNN	Temporal Convolutional Neural Network
VIS	Visible radiometric spectrum

1. Manuscript

Research Article

Comparison of TempCNN and LightGBM for Crop Type Classification using Sentinel-2 Imagery

Frank WILLING[1], Stefan LANG[2*]

[1] Frank Willing; GeoVille GmbH, Sparkassenplatz 2, 6020 Innsbruck

[2] Prof. Dr. Stefan Lang; Department of Geoinformatics – Z_GIS, University of Salzburg, Hellbrunner Str. 34, A-5020 Salzburg, Austria

*Corresponding author. E-mail address: frank.willing@sbg.ac.at

Abstract: High temporal, spectral and spatial resolution Sentinel-2 Satellite Image Time Series (SITS) enable innovative monitoring of vegetation dynamics. Although established classification algorithms have been successfully applied, they are supposed not to make the most of the temporal information. Approaches from the field of Deep Learning (DL) are designed to exploit the unprecedented temporal information more effectively and have already provided promising results. This study aims to evaluate the possibilities and limitations of Temporal Convolutional Neural Network (TempCNN) architectures for crop type classification in Austria. The classification accuracy of the TempCNN is compared to state-of-the-art Machine Learning (ML) classifier Light Gradient Boosting Machine (LightGBM). Initially, LightGBM is applied for crop type grouping and reference classification. Thereafter, both classifiers are evaluated considering heterogeneous classes, the impact of spectral and temporal feature engineering and the ability of spatial generalization. TempCNN did not show the expected improvements. On the contrary, LightGBM generally performs equivalent or better and provides more stable results. Under certain circumstances the complexity of DL architectures does not offer any advantages and therefore established ML classifiers represent a more effective method.

Keywords: remote sensing, time series, crop type classification, Sentinel-2, LightGBM, TempCNN

1.1. Introduction

1.1.1 Satellite Image Time Series Analysis

Against the background of continuing population growth and the consequences of climate change, ensuring food security is becoming an increasingly complex issue. Generating sufficient quantities of food with high nutritional values is one of the key challenges that require a better understanding of the ecosystems involved and smart farming as two fundamental elements (Kamilaris and Prenafeta-Boldú 2018). Crop monitoring is therefore becoming increasingly important in the agricultural sector. Production forecasting and assessing the impacts of crises on food production are valuable information for decision makers. (Matton et al. 2015). In this context crop type classification contributes basic knowledge for more complex procedures like crop area estimation, crop yield forecasting or drought risk analysis on a regional to global scale (Kussul et al. 2016). For the investigation of these critical issues the agricultural sector can vastly benefit from fully free and open remote sensing data. In terms of crop type classification, multitemporal satellite imagery is especially valuable for differentiating between various crop types on the basis of their specific phenological states across the growing season and generally differing spectra (Kussul et al. 2016). The recently available high-resolution Satellite Image Time Series (SITS) from the Sentinel 2 satellites provide an unprecedented source of data especially for land use/cover (LULC) mapping. But the quality and volume of the data also increases applicability in a wide range of non-EO domains. The upcoming opportunities and challenges require new concepts and procedures in processing. This shift in paradigm is also discussed and approached under the term "big Earth data" (Sudmanns et al. 2020). In particular for vegetation and crop mapping tasks high temporal resolution combined with a high spatial resolution sets new standards. A significant increase in classification accuracy of established Machine Learning (ML) algorithms due to further detailed multi-temporal information has already been proven by different experiments (Vuolo

et al. 2018), (Kussul et al. 2016). Although established algorithms provide very good results for the analysis of SITS, they are not able to directly exploit the temporal dimension of the observations (Gómez et al. 2016). Innovative opportunities to use this extended basis of data more effectively might be provided, among others, by approaches from the field of Deep Learning (DL). DL is a powerful methodology from the field of ML often applied in the domains of object recognition or machine translation for instance. This background possibly improves the exploitation of the temporal dimension since certain architectures can also be applied to automatically extract and interpret temporal patterns (Ismail Fawaz et al. 2019). By adding more "depth", in form of multiple layers to models than conventional ML algorithms, those approaches allow the extraction of higher complexity patterns and recently provide promising results exploiting spectral and spatial dimension in the agricultural domain (Kamilaris and Prenafeta-Boldú 2018). One of the main DL architectures generally used is the "Convolutional Neural Network" (CNN) (LeCun and Bengio 1998). As SITS are a key component of the agricultural datasets and the focus is on objects with different phenological development, conventional CNN architectures are not inherently ideally suited and therefore must be modified to allow temporal pattern recognition (Rußwurm and Körner 2017). Out of this necessity "Temporal Convolutional Neural Networks" (TempCNN) were developed as one approach amongst others. Previous approaches applying TempCNN have proven that the architectures exceed current state-of-the-art ML algorithms for SITS classification by automatically extracting temporal features directly from training data (Pelletier et al. 2019). Potential improvements may be expected in terms of feature engineering. The time-consuming and expertise requiring task might be redundant for expert free end-to-end regimes in DL as architectures with a "depth" of more than two hidden layers are supposed to be complex enough to learn temporal feature representations from data more effectively (Zhong et al. 2019). Model stability and better performance on rare and/ or heterogenous crop groups might also be improved as a result of increased model complexity (Pelletier et al. 2019).

Within the pre-processing stage of a classification process, providing and interpreting detailed ground reference data is essential. The most comprehensive reference dataset concerning Austrian agricultural areas is provided by the European Commission's Directorate-General for Agriculture and Rural Development (DG Agriculture and Rural Development) via the Land Parcel Identification System (LPIS). This dataset covers all criteria for institutional administrations, but not yet those required in data science for an adequate classification. LPIS crop classes are too detailed for distinct spectral differentiations and some do not exactly fit the term “crop”.

1.1.2 Aim and Objectives

The milestone in the development of remote sensing represented by the launch of Sentinel 2 requires a rethinking of the conventional approaches to data exploitation. The new paradigms in Earth Observation (EO) emerging from the combination of unprecedented amounts of data and highly performant analysis algorithms must be evaluated. The central aim of this work is to evaluate whether, and if so, how DL techniques make the classification of crops based on SITS analyses more performant. Specifically, the classification results of the established state-of-the-art algorithm Light Gradient Boosting Machine (LightGBM) will be compared to those of an existing TempCNN architecture.

One objective is to identify spectrally or temporally distinguishable crop type classes to make the complex LPIS data applicable for data scientific procedures. Thus, grouping the over 200 classes into 15 to 20 unique and useful groups is a mandatory first operation. Therefore, several experimental classification processes must be executed. Based on the confusion of the respective classes among one another, the crop groups can be further differentiated, which ultimately allows the definition of groups that are as homogeneous as possible. One part of the objective is the application of an appropriate state-of-the-art ML algorithm and to optimize its prediction results based on the respective grouping to provide an appropriate reference model.

This also helps evaluating the dataset in terms of possible features choice, sampling, and crop type grouping.

Based on these reference parameters and results, experiments with the novel TempCNN architecture can be implemented. These experiments include the adjusting of pre-processing steps like increasing temporal interpolation, regrouping of LPIS classes, and changing models training and testing areas. The research questions addressed are listed in the following:

The objective in data pre-processing serves as basis for subsequent comparisons:

- Applying LightGBM algorithm: Which is the most performant contextually useful grouping of classes for Austrian LPIS in 2018?

The general question of improvement:

- Does the TempCNN architecture outperform the LightGBM algorithm?

is analysed considering the following aspects:

- Does the TempCNN achieve higher accuracies in presence of less represented and/or spectrally heterogenous classes?
- Can certain pre-processing steps be omitted using TempCNN? (e.g. Feature engineering)
- Is the TempCNN more stable if the training data source is different from the region to be predicted?

This paper is organized in 5 sections. Following the introduction to the state of research in SITS analysis, section 1.2 describes the Austrian area of interest (AOI), the remote sensing and reference data as well as the classification process used to optimize and compare ML and DL models. It is followed by the presentation, discussion, and conclusion of the respective results.

1.2. Material and Methods

1.2.1 Study Area

The study area is supposed to represent the entire Austrian territory, thus indicating the diversity of the country. To enable effective computational processing, using reduced data volumes, three representative AOI were selected from the regions of Oberösterreich (OÖ), Marchfeld (MF) in Niederösterreich and Steiermark (SM). They are not delimited but named based on regional borders. The AOI are separated into individual processing units for convenient data access and manipulation. The "Single units" (Figure 1) represent individual areas outside the AOI for training and testing.

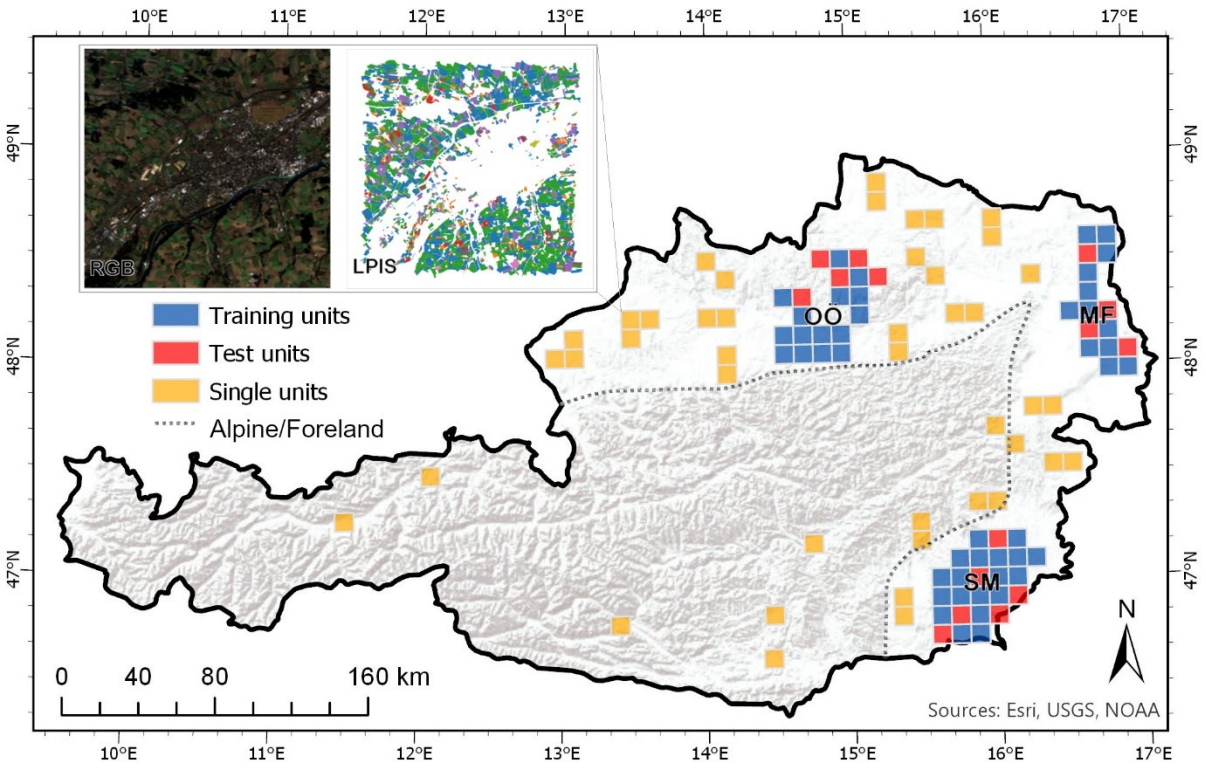


Figure 1: Study area split into train/ test units as well as external single units for spatial generalization checks and examples of LPIS and Sentinel-2 raster data

In the west over 70% of the territory are part of the alpine region. Offering mainly pasture and mowing areas (Figure 2) the alpine region (Figure 1) is mostly unsuitable for cultivation and

excluded from further modelling procedures as it becomes unreasonably complex by varying topographic and agricultural conditions in most areas.

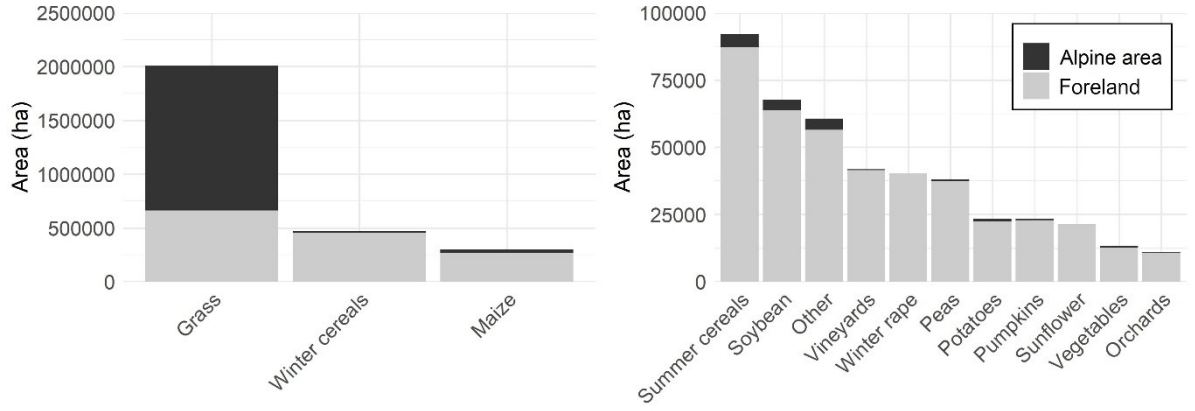


Figure 2: Crop distribution in alpine and foreland Austrian regions

Great plains in the east provide major Austrian agricultural production. Dominated by grasslands, cereals, and maize (Figure 2) the foreland regions (Figure 1) are agriculturally diversely composed and therefore well suited for AOI selection. The conditional diversity of soil, precipitation and cultivation practices results in varying composition of crops amongst the AOIs (Figure 3). While in OÖ mainly the most common crops are represented, SM and particularly MF are composed of a wide cultivation range including minor crops. Differing crop growing stages and cultivation practices challenge models capabilities and allow a detailed performance analysis.

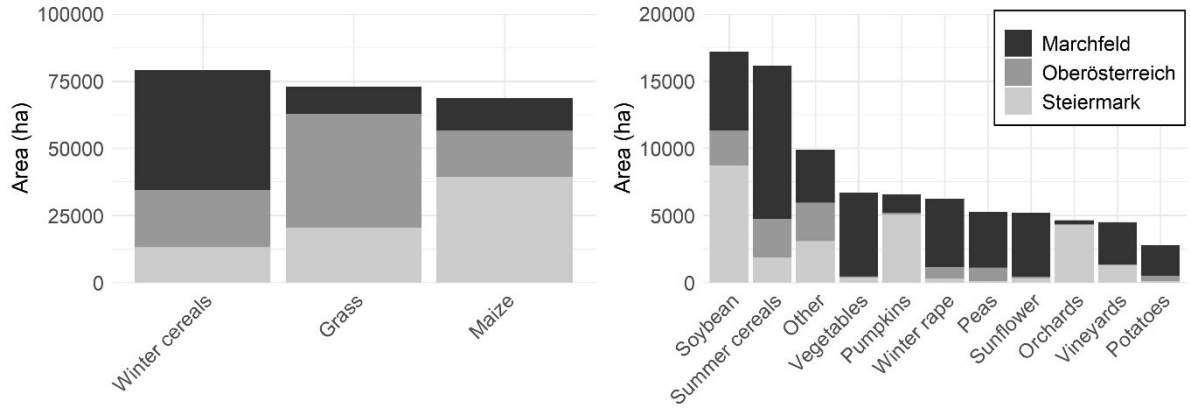


Figure 3: Crop distribution for the 3 regions of the AOI

1.2.2 Data

1.2.2.1 Sentinel-2 Imagery

The remotely sensed input dataset is composed of Sentinel 2 surface reflectance, available as pre-processed "bottom-of-atmosphere" Level 2A product (see RGB inset in Figure 1) acquired and provided under the Copernicus project initiated by the European Space Agency and the European Commission. The data is derived from the Multispectral Instrument (MSI) on the satellites "S2B" and "S2A". The optical instrument samples 12 multispectral bands: four bands at 10m (Bands 2, 3, 4, 8), six bands at 20m (Bands 5, 6, 7, 8a, 11, 12), three bands at 60m (Bands 1, 9, 10) spatial resolution (ESA). Bands B02, B03, B04, B05, B06, B07, B08, B8A, B11, B12 are potentially relevant for vegetation monitoring and therefore used as features. On their basis Normalized Difference Index (NDVI), Normalized Water Index (NDWI) and Normalized Euclidean Distance (NORM) are calculated as additional features in the context of this work. According to (Pelletier et al. 2016) additional pre-calculation of temporal features considering phenological stages is not reasonable for state-of-the-art ML approaches. The combination of two satellites enables a high revisit frequency up to 5 days. The temporal resolution potentially provides 54 images for the investigation period from 01.01.2018 to 30.09.2018. But the actual SITS is reduced mostly by winterly cloud coverage as it includes

only timeframes with a valid data fraction over 70%. The limiting overall intersecting dates are 13.04.2018 to 28.09.2018. To overcome technical complications by missing data and varying length of SITS, they are resampled and interpolated on 8-day interval. Thus, the generated SITS have a consistent length of 21 equidistant timeframes.

1.2.2.2 Ground Reference Data

The LPIS is a geographical information system operated by several European member states. It records the geographical position and spatial extent of agricultural areas on reference parcel level based on orthophotos (Počivavšek and Ljuša 2013). Austrian LPIS for 2018 includes 1.2.522.622 reference parcel polygons (see LPIS inset in Figure 1) covering an area of about 3.209.479 hectares. To ensure an appropriately detailed database for administration, the institutions distinguish between more than 200 different crop type classes.

1.2.2.3 Dataset Partition

For modelling the dataset is split into training and test sets. The training set is used to train the individual classification algorithms. The final classification results are evaluated using the test set. Each AOI is further divided into smaller processing units. To ensure pixel for training and testing originate from different agricultural fields, about 75% of these units are assigned to training and about 25% to test set (Figure 1). The random sampling of pixel is executed on processing unit level. Only crop groups represented by at least 1000 pixel are included in the respective sampling process with a share of 500 pixel. As a result, there are 203.230 pixel in training and 74.044 in test set.

1.2.3 Classification

1.2.3.1 Benchmark ML Classifier

The LightGBM algorithm (Guolin Ke et al. 2017) is applied as reference classifier for crop type grouping and accuracy benchmarking, representing one of today's most popular non-DL algorithms. Amongst others (Ustuner and Balik Sanli 2019) received strong results applying

LightGBM for crop type classification, and (Krishna Moorthy et al. 2019) recently found it outperforming state-of-the-art algorithms Random Forest (RF) (Breiman 2001) and XGBoost (Chen and Guestrin 2016) for leaf and wood classification on radar data. Like the latter, LightGBM is an implementation of Gradient Boosting Decision Tree (GBDT) algorithm. Introduced by Microsoft it is supposed to optimise GBDTs computational bottlenecks. Unique applications, like leaf-wise tree growing for maximising loss reduction, Gradient-based One-Side Sampling (GOSS) for increasing information gain and Exclusive Feature Bundling (EFB) for reducing training complexity (Guolin Ke et al. 2017), enable benchmarking results. The architecture is implemented using Pythons Scikit-learn package (Scikit-learn homepage, https://scikit-learn.org). Hyperparameters are applied following the default values of the corresponding package (Microsoft Corporation).

1.2.3.2 DL Classifier

TempCNN extend the idea of automatic feature extraction behind common CNN. Convolutional layers reduce features complexity and highlight patterns (LeCun and Bengio 1998). The crux of applying an CNN architecture for the exploitation of SITS sequential data is to implement one-dimensional convolutional filters capable of capturing temporal patterns. In that context (Pelletier et al. 2019) successfully developed an effective arrangement of architectural components and hyperparameters in extensive studies.

According to their results the subsequently applied architecture is composed of three convolutional layers including 64 units, one dense layer including 256 units and one SoftMax layer. The filter size is set to 5 and the dropout rate to 0.5. No pooling layers are implemented. Adam optimization with standard parameters and a batch size of 32 is used for training with number of epochs set to 10. The architecture is built using the Keras library (Keras homepage, https://keras.io) on top of Tensorflow (TensorFlow homepage, https://www.tensorflow.org) and the model is trained and evaluated using Scikit-learn package again.

1.2.3.3 Evaluation

Four research questions were formulated to allow an overall assessment. Each included the computation of matrices and overall accuracies to evaluate the performance of the classifiers on the test set. Considering overall accuracy only in this case is adequate as the threshold-bound sampling strategy guarantees a balanced dataset (Guo et al. 2008).

Beginning with considerably grouping LPIS classes, a first grouping based on functional preferences was introduced. From there, successive regrouping and synchronous evaluation of LightGBM classification results was required. For the subsequent comparison of the models, the focus first was on the overall accuracy achieved for individual heterogeneous groups. Second, the individual performances were evaluated with respect to feature engineering. The algorithms were applied on data based on either 4-day or 8-day temporal resampling range and on datasets both including and missing precalculated spectral features. At last the spatial stability was examined. The models were applied on data from single units (Figure 1) outside the training area sampled across the entire Austrian territory.

1.3. Results

1.3.1 Crop Grouping

Step 1 introduces the group Multi use which includes LPIS classes that do not define one main crop. It is not intended primarily to increase the model's accuracy, but to clearly separate crop groups contextually. As expected, there is no significant effect on the accuracy of the groups (Figure 4). In contrast, step 2 represents a significant process. At least for Grass it allows the accuracy to increase about 66 % (Figure 4) as it combines the grassy classes Leafy Legumes and/ or Grass mixture, Pasture, (Alpine-) Meadows and Grass in the latter. The strong increase improves overall accuracy about 6%. The following two steps also combine groups. The groups Poppy and Soft fruits are performing poorly because of less representation and spectral diversity. Therefore, they are assigned to Other in step 3. The groups Beets, Fallow land, Hop,

Buckwheat are even less represented and therefore not listed in figure 4. Together with Beans they are also assigned to Other in step 4. The final step 5 allows Grass to improve about 13% and thereby increases overall accuracy to about 77% (Figure 4). Here, the grassy LPIS classes: Once per year mow meadow ("Einmähdige Wiese"), Crop rotation- natural vegetation without planted vegetation ("Grünbrache"), Crop rotation - non cultivated for some time ("Grünlandbrache"), Different green areas ("Sonstige Grünlandflächen") and Changing meadow ("Wechselwiese") are assigned to Grass. In total, the overall accuracy increases by about 9% from 68% to 77%.

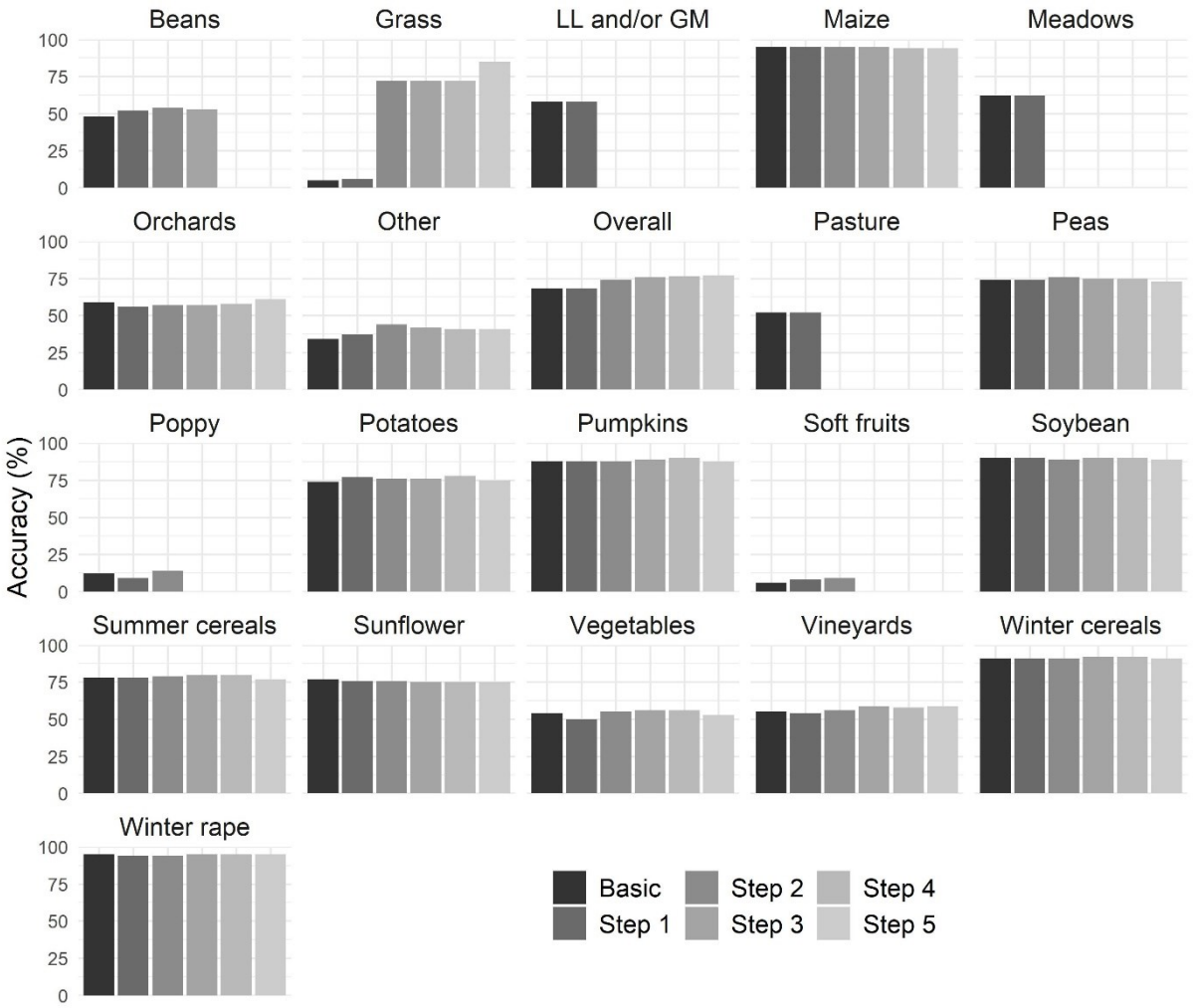


Figure 4: Stepwise accuracy check for single crop grouping steps related to individual crop groups

1.3.2 Model Stability on heterogenous Groups

Heterogeneity appears in different patterns amongst the groups (Figure 5). Other and Vegetables consist of numerous, spectrally varying classes. Between the actual crops of Orchards and Vineyards large grassy areas are common and Grass, Winter- and Summer cereals include classes that are subject to varying cultivation practices. In the context of the first potential heterogeneity category the TempCNN model slightly outperforms the LightGBM model for Vegetables about 4%, while the latter clearly outperforms the former for Other about 13%. Similar results occur for the second category. LightGBM slightly outperforms TempCNN for Orchards about 4% and for Vineyards it is the other way around with a difference of 8 %. In the third category both models perform similarely, LightGBM slightly outperforming TempCNN for Grass about 3% and Summer cereals about 5%.

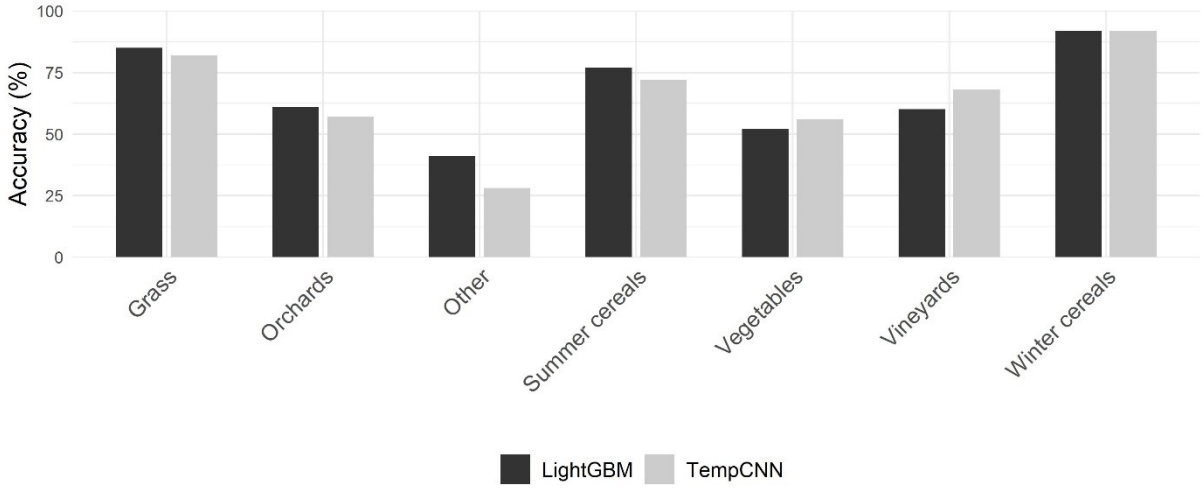


Figure 5: LightGBM and TempCNN performances on heterogenous groups

1.3.3 The Impact of Feature Engineering – Indices

In general, both models perform similarely on datasets that do include spectral indices and those that do not (Figure 6). With the focus on specific crop groups, LightGBM shows only little differences. Vegetables and Vineyards are the only groups with noticeable discrepancies, that are still less than 5% in accuracy. In both cases LightGBM performs slightly worse on datasets

missing indices. For TempCNN classification results are more varying. Regarding the dataset missing indices, remarkable differences occur for Vineyards where the model performs about 6% worse, Winter cereals with a decrease of 7% and a decrease of 8% for Sunflower. Remarkably increasing accuracy of 11% the model achieves on Other.

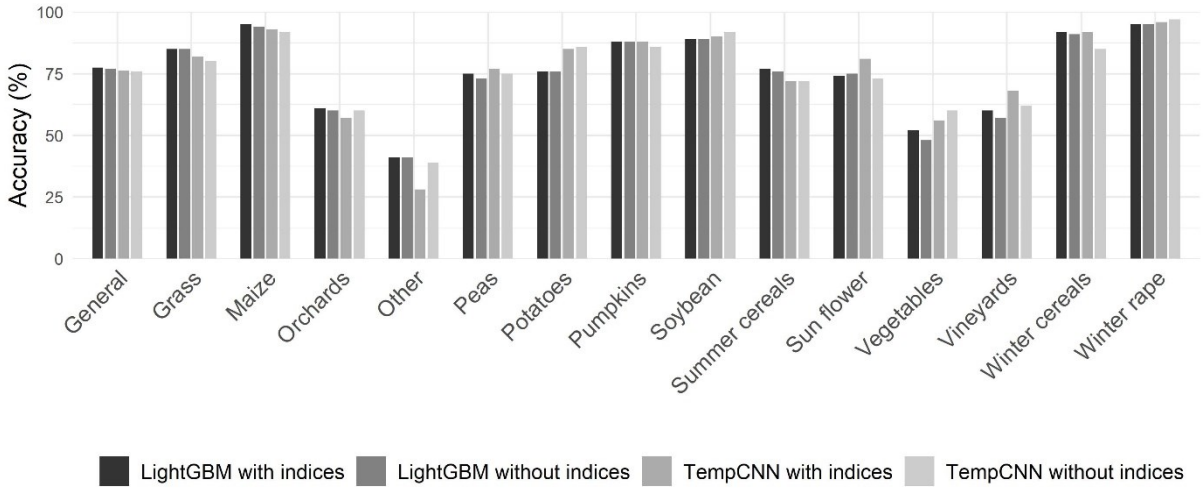


Figure 6: LightGBM and TempCNN performances on datasets including and missing spectral indices

1.3.4 The Impact of Feature Engineering – temporal Resolution

Again, both models perform similarely on 4- and 8-day resampling in general. Increased temporal resolution improves LightGBM accuracy about 1% and TempCNN accuracy about 2% (Figure 7). Across all groups LightGBM performs very consistently. Only unremarkable differences occur for Other and Vegetables. In contrary, TempCNN classification results vary significantly. It performs remarkably better on the dataset with increased temporal resolution about 6% for Vegetables and about 10% for Other. For Potatoes, its results are about 25% worse.

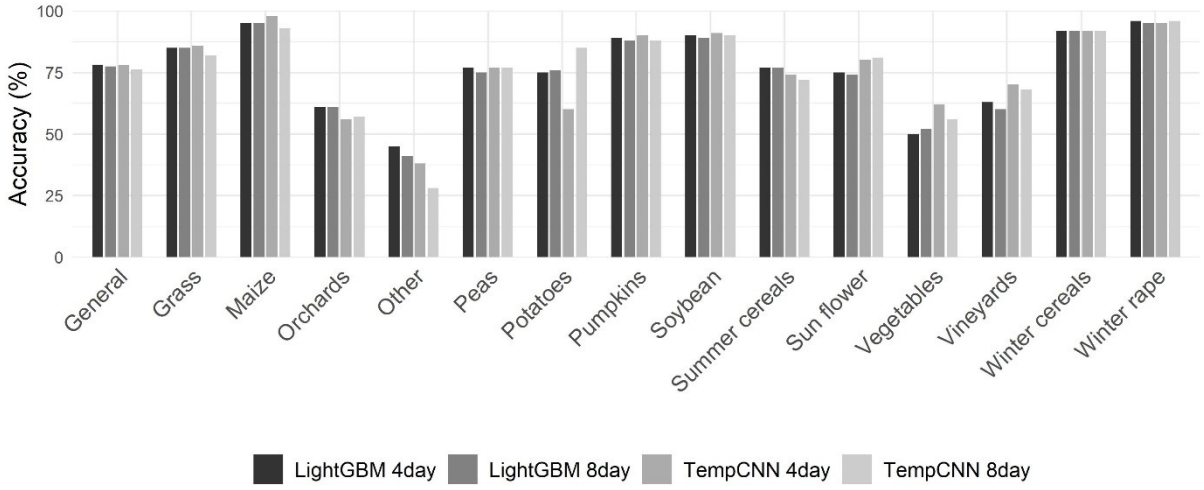


Figure 7: LightGBM and TempCNN performances on 4- and 8-day resampled datasets

1.3.5 Model Stability on spatial Variation

Figure 8 shows the mean accuracy for both models performing on data from single units (Figure 1). It displays significant differences for certain classes. LightGBM performs significantly better on Grass about 6% and on Winter cereals about 25%. However, its results are worse about 6% for Pumpkins, about 12% for Vineyards, about 13% for Vegetables and about 17% for Sunflower.

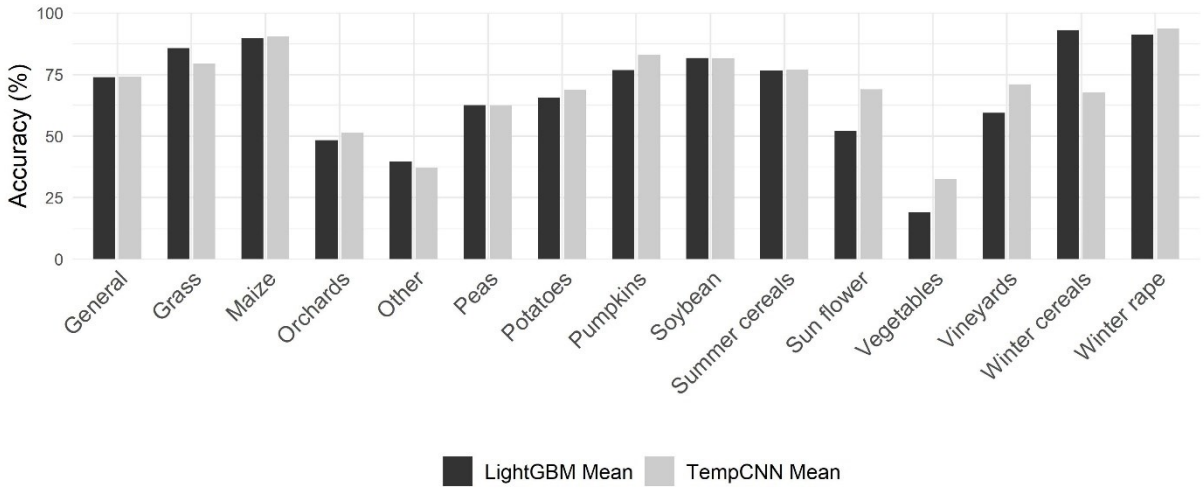


Figure 8: LightGBM and TempCNN performance on test data from outside the training area

An examination of the results for individual classes inside each crop group reveals some interesting patterns. For Potatoes LightGBM outperforms TempCNN with over 8% at times in areas dominated by potatoes for industrial production ("Stärkeindustriekartoffeln") which is not represented well in the training dataset (Figure 9). For Maize, the pattern is not directly crop class related. Figure 10 shows LightGBM generally outperforming TempCNN about 2% to 8% in alpine regions dominated by Silage Maize ("Silomais"). In foreland areas both models perform either similarely or TempCNN outperforms LightGBM regardless of class distribution about 2% to 8%.

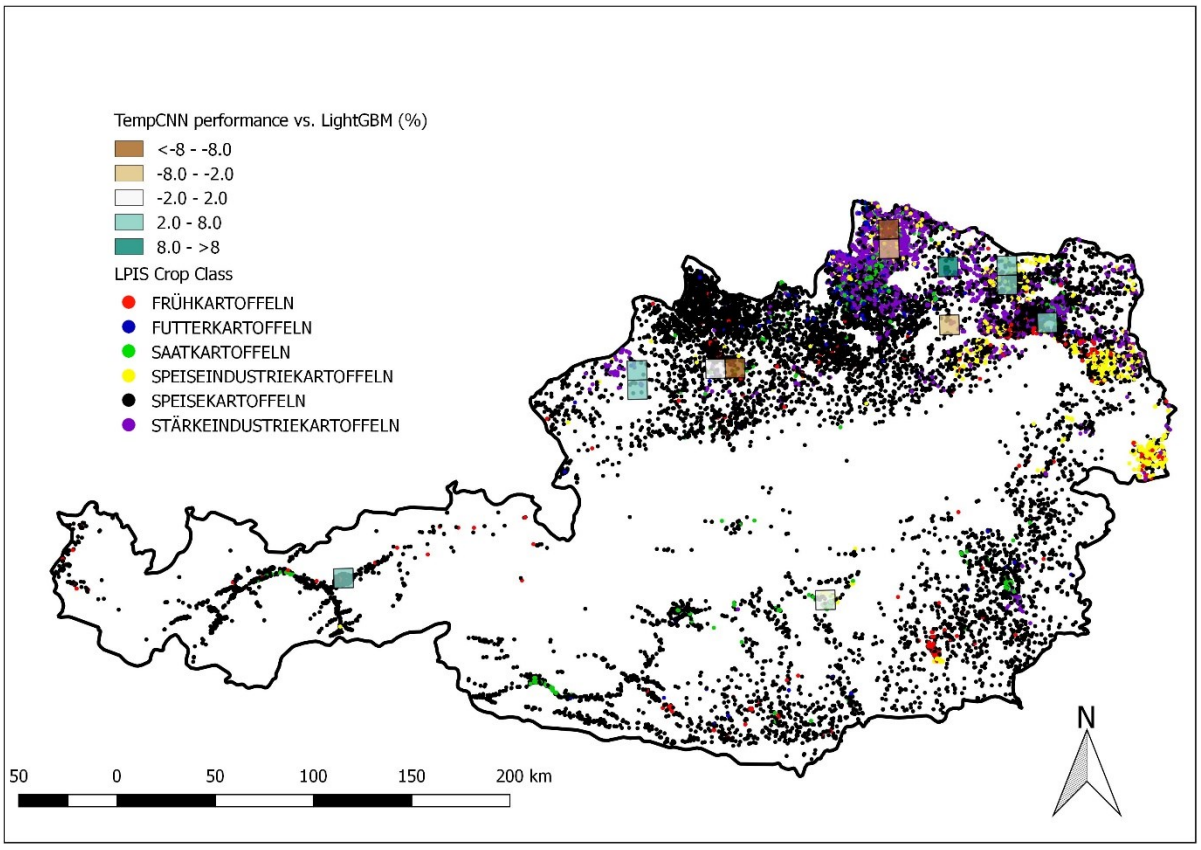


Figure 9: Distribution of LPIS crop class Potatoes and respective TempCNN performance in contrast to LightGBM

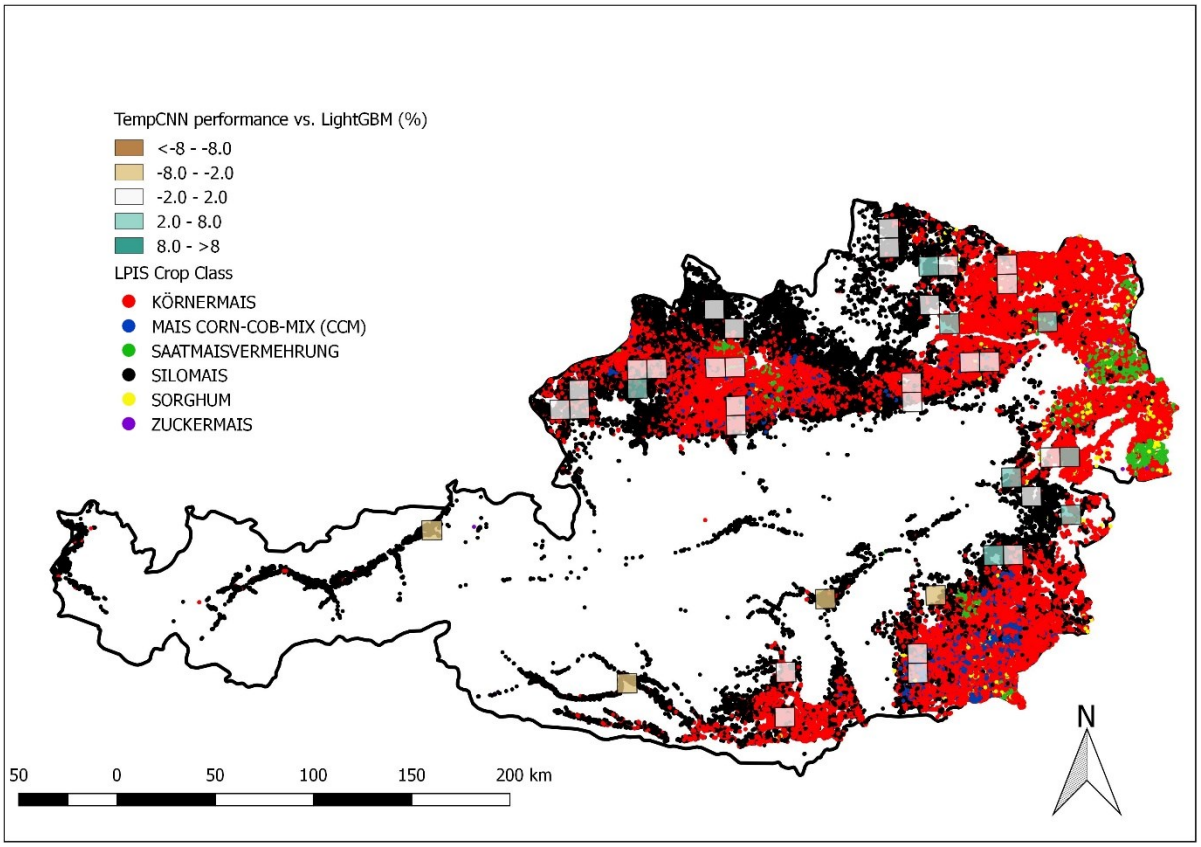


Figure 10: Distribution of LPIS crop class Maize and respective TempCNN performance in contrast to LightGBM

1.4. Discussion

Studies of (Pelletier et al. 2019) and (Zhong et al. 2019) show that CNN's efficiency in domains like image and text recognition with an existing relationship between the dimensions is transferable to the sequential relationship of multi-temporal remote sensing observations. A more detailed examination of these findings is required, regarding the results of the previous section.

For the success of the crop type grouping presented in section 1.3.1 the increasing accuracy of Grass had the most relevant impact on overall accuracy. The grassy classes are grouped as they are very similar in terms of cultivation practices, resulting in common temporal and spectral characteristics. The overall accuracy might be further improved, by e.g. not individually considering less performant groups like Orchards, Vegetables or Vineyards. However, for the

essential part of model comparison, a diverse grouping including also weakly performing groups is beneficial. Furthermore, already well performing classes did not change much during grouping and could not be improved. To enforce a more balanced evaluation both algorithms could be considered for underlying crop grouping. As the crop groupings were only investigated using the LightGBM model, further evaluation should consider the effect of different croup groupings on TempCNN results.

The subsequent experiments do not support the introductory thesis of superior TempCNN, as heterogenous groups in section 1.3.2 could not be separated more precisely by increased model complexity of TempCNN in neither of the categories. Considering overall accuracy, TempCNN cannot benefit from automatic feature extraction. Neither skipping indices pre-calculation in section 1.3.3 nor enhancing temporal resolution via increased resampling in section 1.3.4 lead to outperforming results. These discrepancies with the literature may originate from extensive pre-processing, preceding the actual modelling process of this study. For technical reasons, it is necessary to provide the algorithms with data that is equidistant or at least identical in extent. As stated in section 1.2.2.1 for a study area covering the size of the Austrian territory, this is only possible by temporal resampling and interpolation. Through this generalization the dataset loses complexity and informational quality, which possibly prevents TempCNN from reaching its maximum potential. Preserving maximum dataset complexity by using all available observations and only interpolating cloud cover like (Zhong et al. 2019) should be subject of further studies. However, it is to be expected that the variety of cloud cover conditions caused by the vast extension of the study area is likely to cause significant confusion on spectro-temporal signatures of crop type classes.

Using both models for classification of data from single units outside the training area in section 1.3.5 raises multiple questions and therefore provides some basis for further comprehensive studies. TempCNN outperforming results for Grass and Winter cereals and the inverted

outcome for Pumpkins, Vineyards, Vegetables and Sunflower indicate that TempCNN is more stable on less represented groups, whereas LightGBM is more stable on the mainly represented groups. However, for the formulation of a general statement further investigation is required.

Although the outcome of the detailed examination of individual crop groups in section 1.3.5 is not quantifiable and originates from visual interpretation, it is worth mentioning and provides indications for further studies. Different potato types, especially high maintenance Seed Potatoes ("Saatkartoffeln") are managed in different ways. The results for Potatoes suggest that TempCNN struggles faced with these differing cultivation practices and is less capable of identifying those crop types that are not common in the training data. Patterns are also visible for Maize which has higher climatic requirements than Winter cereals, for example (Sinabell et al. 2014). These requirements can lead to temporally differing growth stages in Alpine and Foreland regions because of climatic differences. The outcome especially for Silage Maize ("Silomais") suggests that TempCNN is not able to generalize these differences in growth patterns well. Again, further investigations for the formulation of a generally valid statement should be subject to future scientific research.

1.5. Conclusion

In this study the state-of-the-art ML algorithm LightGBM and a recently promisingly performing DL TempCNN architecture were applied for crop type classification on Sentinel-2 SITS. The referring literature provokes the expectation that TempCNN can generally outperform LightGBM. The results received in the course of this work, however, show that the assumption must be further differentiated. In general, TempCNN was neither able to better separate spectrally heterogeneous groups, nor was it possible to detect advantages in abandoning the pre-calculation of spectral features or the increase of the temporal resolution. On the contrary, LightGBM provided more stable results regarding individual groups. Also

tests on data from outside the training area suggest that TempCNN is less able to generalize group specific spectro-temporal signatures compared to LightGBM.

The experimental results of this work show that LightGBM provides better outcomes based on an extensively pre-processed and therefore generalized and less complex dataset. Research from corresponding literature leads to the assumption that TempCNN achieves better results for less prepared and therefore more complex datasets. The use of such data would reduce the dataset related workload. However, it is to be expected that such conclusions only apply to study areas with a limited extent or consistent cloud cover conditions. For a study area the size of Austria, as in this study, the improved practicability of DL applications is questionable. To obtain consistent and thus identifiable spectro-temporal signatures for crop classes inside the respective groups, more extensive pre-processing is necessary. The conclusion of this thesis is therefore that in the case of extensive and diverse study areas and the associated exhaustive data preparation, both approaches achieve satisfactory results, but the LightGBM algorithm represents the classifier of choice. Future research could investigate whether the conclusions drawn are transferable to other large-scale areas with different crop classes and cultivation practices. The impact of data pre-processing should also be further evaluated to determine an appropriate ratio between workload and classification accuracy.

Acknowledgment

I would like to thank the management team of GeoVille GmbH for giving me the opportunity to write my master's thesis embedded in the Perceptive Sentinel project and thus in a professional framework on an exciting and future-oriented topic. Especially David Kolitzus and Samuel Barret supported me with a lot of helpful feedback, discussions, and suggestions.

Many thanks to the whole UNIGIS team and my supervisor Prof. Dr. Stefan Lang for the competent support.

Finally, I would like to thank my family and friends for the patience and support they have shown during my studies.

References

Breiman, L. (2001): Random Forests. In: *Machine Learning* 45 (1), S. 5–32. DOI: 10.1023/a:1010933404324.

Chen, Tianqi; Guestrin, Carlos (2016): XGBoost. In: Balaji Krishnapuram, Mohak Shah, Alex Smola, Charu Aggarwal, Dou Shen and Rajeev Rastogi (Hg.): Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. the 22nd ACM SIGKDD International Conference. San Francisco, California, USA, 13.08.2016 - 17.08.2016. New York, New York, USA: ACM Press, S. 785–794. DOI: 10.1145/2939672.2939785.

ESA: Sentinel-2 Handbook. URL: https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook, Last accessed: 27.08.2020.

Gómez, Cristina; White, Joanne C.; Wulder, Michael A. (2016): Optical remotely sensed time series data for land cover classification: A review. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 116, S. 55–72. DOI: 10.1016/j.isprsjprs.2016.03.008.

Guo, Xinjian; Yin, Yilong; Dong, Cailing; Yang, Gongping; Zhou, Guangtong (2008): On the Class Imbalance Problem. In: Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (Hg.): On the Class Imbalance Problem. Fourth International Conference on Natural Computation. Jinan, Shandong, China: IEEE, S. 192–201. DOI: 10.1109/ICNC.2008.871.

Guolin Ke; Qi Meng; Thomas Finley; Taifeng Wang; Wei Chen; Weidong Ma et al. (2017): LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: *Advances in neural information processing systems*, S. 3146–3154.

Ismail Fawaz, Hassan; Forestier, Germain; Weber, Jonathan; Idoumghar, Lhassane; Muller, Pierre-Alain (2019): Deep learning for time series classification: a review. In: *Data Min Knowl Disc* 33 (4), S. 917–963. DOI: 10.1007/s10618-019-00619-1.

Kamilaris, Andreas; Prenafeta-Boldú, Francesc X. (2018): Deep learning in agriculture: A survey. In: *Computers and Electronics in Agriculture* 147, S. 70–90. DOI: 10.1016/j.compag.2018.02.016. DOI: 10.1016/j.compag.2018.02.016.

Krishna Moorthy, Sruthi M.; Calders, Kim; Vicari, Matheus B.; Verbeeck, Hans (2019): Improved Supervised Learning-Based Approach for Leaf and Wood Classification from LiDAR Point Clouds of Forests. In: *IEEE Trans. Geosci. Remote Sensing*, S. 1–14. DOI: 10.1109/TGRS.2019.2947198.

Kussul, Nataliia; Lemoine, Guido; Gallego, Francisco Javier; Skakun, Sergii V.; Lavreniuk, Mykola; Shelestov, Andrii Yu. (2016): Parcel-Based Crop Classification in Ukraine Using Landsat-8 Data and Sentinel-1A Data. In: *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing* 9 (6), S. 2500–2508. DOI: 10.1109/JSTARS.2016.2560141.

LeCun, Yann; Bengio, Yoshua (1998): Convolutional networks for images, speech,and time-series. In: *The Handbook of Brain Theory and Neural Networks*, S. 255–258.

Matton, Nicolas; Canto, Guadalupe; Waldner, François; Valero, Silvia; Morin, David; Inglada, Jordi et al. (2015): An Automated Method for Annual Cropland Mapping along the Season for Various Globally-Distributed Agrosystems Using High Spatial and Temporal

Resolution Time Series. In: *Remote Sensing* 7 (10), S. 13208–13232. DOI: 10.3390/rs71013208.

Microsoft Corporation: LightGBM Parameters. URL: https://lightgbm.readthedocs.io/en/latest/Parameters.html, Last accessed: 27.08.2020.

Pelletier, Charlotte; Valero, Silvia; Inglada, Jordi; Champion, Nicolas; Dedieu, Gérard (2016): Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas. In: *Remote Sensing of Environment* 187, S. 156–168. DOI: 10.1016/j.rse.2016.10.010.

Pelletier, Charlotte; Webb, Geoffrey; Petitjean, François (2019): Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series. In: *Remote Sensing* 11 (5), S. 523-548. DOI: 10.3390/rs11050523.

Počivavšek, G.; Ljuša, M. (2013): Characteristics of the Land Parcel Identification System (LPIS) as the main subcomponent of the agriculture information system. In: *Ege Üniversitesi Ziraat Fakültesi Dergisi* (Special Issue Volume 1), S. 133–138.

Rußwurm, M.; Körner, M. (2017): Multi-Temporal Land Cover Classification with Long Short-Term Memory Neural Networks. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-1/W1, S. 551–558. DOI: 10.5194/isprs-archives-XLII-1-W1-551-2017.

Sinabell, F.; Kratena, K.; Sommer, M.; Kappert, R.; Kaul, H. (2014): Maisanbau in Österreich. Ökonomische Bedeutung and pflanzenbauliche Herausforderungen. WIFO. Vienna.

Sudmanns, M.; Tiede, D.; Lang, S.; Bergstedt, H.; Trost, G.; Augustin, H.; Baraldi, A.; Blaschke, T. (2020): Big Earth data: disruptive changes in Earth observation data management and analysis? In: *International Journal of Digital Earth* 13 (7), S. 832-850. DOI: 10.1080/17538947.2019.1585976.

Ustuner, Mustafa; Balik Sanli, Fusun (2019): Polarimetric Target Decompositions and Light Gradient Boosting Machine for Crop Classification: A Comparative Evaluation. In: *IJGI* 8 (2), S. 97-112. DOI: 10.3390/ijgi8020097.

Vuolo, Francesco; Neuwirth, Martin; Immitzer, Markus; Atzberger, Clement; Ng, Wai-Tim (2018): How much does multi-temporal Sentinel-2 data improve crop type classification? In: *International Journal of Applied Earth Observation and Geoinformation* 72, S. 122–130. DOI: 10.1016/j.jag.2018.06.007.

Zhong, Liheng; Hu, Lina; Zhou, Hang (2019): Deep learning based multi-temporal crop classification. In: *Remote Sensing of Environment* 221, S. 430–443. DOI: 10.1016/j.rse.2018.11.032.

2. Report

In the technical Report, the methods applied for answering the research questions raised and addressed in the preceding manuscript part of this thesis are outlined. These include an introduction of the software used as well as a presentation of the processing workflow from data acquisition towards pre-processing and classification up to the final accuracy assessment.

2.1 Software

Introducing the report, in this chapter an overview of important software used is presented. It forms the framework and is mainly used to access and pre-process a dataset suitable for developing, training, and evaluating the respective models. Most of the tools are well known, especially in the remote sensing community. Still, to present a complete documentation of the method applied a short introduction is given.

2.1.1 Python

Python is a popular and powerful interpreted programming language distributed by the Python Software Foundation. It was first released in 1991. The current Python 3.7 version, which is used in the context of this thesis, was released in 2018. Python is a complete language and platform that can be used for both research and development and developing production systems. Due to this and the huge amount of external libraries python is a widely used language in many scientific projects. It is also one of the most popular languages in the fields of ML as it offers many powerful ML and especially DL libraries.

- **Scikit-learn**: Scikit-learn is a Python module for ML built on top of the SciPy module. The development is focused on providing a solid implementation for ML that focuses on the essential functions. A clean, consistent, and simple API is what makes the library stand out. Once the principles of applying Scikit-learn to a model are understood, it is relatively easy to apply it to other models.

- **TensorFlow**: TensorFlow is a powerful open source Python library for implementing and deploying large-scale machine learning models in recent years it became one of the most popular libraries for deep learning.
- **Keras**: Keras is a deep learning framework for Python that provides a convenient way for defining and training models. Amongst others, it is capable to run on top of TensorFlow, enabling fast experimentation.

Furthermore, Python offers a great range of other libraries useful for data pre-processing. Some of those applied in this work are:

- **Numpy**: Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.
- **Pandas**: Pandas is, amongst others, based on numpy-functions and provides special functions and data structures for the manipulation of numerical tables and SITS. Data is stored in pandas.Series (instance) and pandas.DataFrames (instances stored as tabular data)
- **GeoPandas**: GeoPandas adds support for geographic data to Pandas objects. It implements geopandas.GeoSeries and geopandas.GeoDataFrames which are subclasses of pandas.Series and pandas.DataFrame respectively. GeoPandas objects can act on shapely geometry objects and perform geometric operations.

2.1.2 eo-learn Library

The eo-learn library requires a detailed introduction. As the main tool for data acquisition and pre-processing it serves as interface for ML analysis and interpretation of the results. The open-source framework for remote sensing data analysis was developed and published in 2018 by Sinergise (Sinergise homepage, http://www.sinergise.com). It was developed under the

Perceptive Sentinel European grant and therefore received funding from European Union's Horizon 2020 Research and Innovation Programme. GeoVille Information Systems and Data Processing GmbH acts as a partner in this project. The term EO covers the collection of data, e.g. images, about our planet via satellite remote-sensing technologies. This provides scientists and decision makers with valuable information to better understand our environment. Between the acquisition of a satellite image and an actionable information, however, there is a large processing effort. eo-learn as a collection of modular Python sub-packages allows easy and quick processing of spatio-temporal data to prototype, build and automate these required large scale EO workflows for AOIs of any size. It also directly enables the application of state-of-the-art tools for computer vision, ML and DL packages in Python to the data. Especially for non-experts to the field of remote sensing and ML it makes extraction of valuable information from satellite imagery easier and more comfortable. Time consuming research and downloading a large amount of data as holding it available is replaced by simple access to the online database SentinelHub containing data of the Copernicus and Landsat programs. Therefore, the functionalities of the eo-learn library often make use of interfaces of the SentinelHub Python package. This package allows users to make OGC (WMS and WCS) web requests to download and process satellite images within custom Python scripts. It supports Sentinel-2 L1C and L2A, Sentinel-1, Landsat 8, MODIS and DEM data source (SentinelHub repository, https://github.com/sentinel-hub/sentinelhub-py). The idea of eo-learn is to implement e.g. complete classification workflows from the download of the data to the evaluation of the results in one eo-learn based workflow.

The core modules of the framework are EOPatches, EOTasks and EOWorkflows. EOPatches store multi-temporal imaging and non-imaging data in the format of NumPy arrays and Shapely polygons (Table 1).

Table 1: EOPatch data formats

Character	***Name***	***Content***
Time-dependend	DATA	spatio-temporal raster data (e.g. bands)
	MASK	spatio-temporal raster masks (e.g. cloud mask)
	VECTOR	spatio-temporal raster data (e.g. parcels)
	SCALAR	temporal float-values (e.g. cloud coverage)
	LABEL	temporal int-values (e.g. classification label)
Time-independend	DATA_TIMELESS	spatial raster data (e.g. digital elevation model)
	MASK_TIMELESS	spatial raster mask (e.g. crop type labels)
	VECTOR_TIMELESS	spatial vector data (e.g. super-pixels)
	SCALAR_TIMELESS	float-scalar values (e.g. probabilities)
	LABEL_TIMELESS	int-scalar values (e.g. processing flags)
Meta-data	BBOX	bounding box of the AOI in a given Coordinate Reference System
	TIMESTAMP	list of datetime-objects for each frame in the time-series
	META_INFO	dictionary of meta-info pertaining to the time-series (e.g. OGC request parameters)

EOTasks are easy to implement sub packages for performing specific operations on EOPatch instances. They are subdivided into the groups eo-learn-core, - io, -mask, -features, -geometry, -ml-tools and -coregistration also listed in table 2.

Table 2: eo-learn subpackages

Sub package (EOTask)	***Functions***
eo-learn-core	The main sub package which implements basic building blocks (EOPatch, EOTask and EOWorkflow) and commonly used functionalities
eo-learn-io	Input/output sub package that deals with obtaining data from Sentinel Hub services or saving and loading data locally
eo-learn-mask	The sub package used for masking of data and calculation of cloud masks
eo-learn-features	A collection of utilities for extracting data properties and feature manipulation
eo-learn-geometry	Geometry sub package used for geometric transformation and conversion between vector and raster data
eo-learn-ml-tools	Various tools that can be used before or after the ML process
eo-learn-coregistration	The sub package that deals with image co-registraion

An EOWorkflow can be described as acyclic graph of EOTasks that form complete EO processing pipelines including logging and monitoring.

Based on the eo-learn library Sinergise builds up the Perceptive Sentinel platform. It is supposed to be an intermediate EO service for fast, efficient, and easy design, exposure and exploitation of EO-processing chains based on multi-temporal and multispectral EO and non-EO data (Perceptive Sentinel homepage, http://www.perceptivesentinel.eu/). In the context of this project Sinergise created an open-source ML pipeline for LULC classification at a country-level. The pipeline is meant to be applied to any country (Sinergise LULC example, https://github.com/sentinel-hub/eo-learn/blob/master/examples/land-cover-map/SI_LULC_pipeline.ipynb). The Workflow presented in the following chapters is oriented on that example workflow.

2.2 Data Pre-Processing

In the following an explanation of the workflow for data pre-processing is given. This includes accessing remote sensing and reference data, feature engineering, interpolation, and sampling.

2.2.1 AOI

Sentinel-2 products are generally available in the form of 100x100 km tiles. The eo-learn library enables the download for custom areas, namely EOPatches. In this specific use case, the total 8,822 million km² of Austrian territory is divided into 991 EOPatches, each measuring 100 km². For the three representative regions, 61 of these 991 Patches are selected. 20 of these are in OÖ, 16 in MF and 25 in SM.

2.2.2 EO-Data

The eo-learn library is used to download the L2A products. The task accesses SentinelHub's Web Coverage Service and offers the advantages of a precise download application. Instead of downloading imagery from SciHub, using the JP2 format, processing, re-projecting, or

mosaicing the data (Kamilaris and Prenafeta-Boldú 2018), it can be integrated into the workflow as a numpy array right away. Also, compared to conventional procedures less storage volume and processing power is required. For the three regions the bands B02, B03, B04, B05, B06, B07, B08, B8A, B11 and B12 are downloaded and if necessary, converted to a resolution of 10x10 m per pixel. Thus, from the 12 available bands, all 10 bands relevant for the observation of the earth's surface are included into the dataset. Even bands which initially appear to have little value for the classification can provide valuable information for the modelling process. The bands B01, B09 and B10 are available in 60 m resolution which causes a high blur radius. In consequence these are prone to the mixed pixel problem, which originates from various land cover types in a single pixel area and reduces the reliability of the classification results. In addition, they are mostly used for atmospheric image correction and since the data is already of corrected Level-2A quality they tend to be redundant. For the download only those sentinel tiles are considered, which are covered with clouds up to 80 %. The corresponding period is defined from 01.01.2018 to 30.09.2018. The last three month of the year are ignored as amongst others (Maponya et al. 2020) state, that more accurate results are achieved using only beneficial input data acquired during peak growth stages. Within this time range all images which are more than 2 hours apart should be recorded as single timeframes.

Despite high temporal and multispectral resolution, Sentinel datasets suffer a lack of consistency due to clouds and cloud artifacts. These common disruptions are frequently responsible for subtractions in the classification accuracy and therefore need to be detected and excluded. One of the actions for detecting invalid pixel is adding a Scene Classification Map (SCL) produced by Sen2Cor (ESA Plugins). Sen2Cor is a processor for Sentinel-2 Level 2A product and formatting. It performs the atmospheric-, terrain and cirrus correction of Top- or Bottom-Of-Atmosphere Level 1C input data. Also, it processes Aerosol Optical Thickness-,

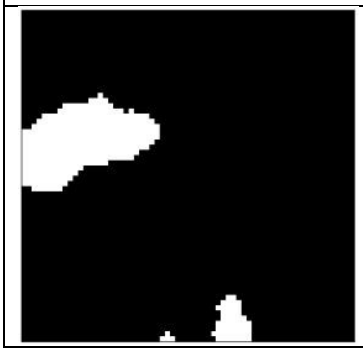
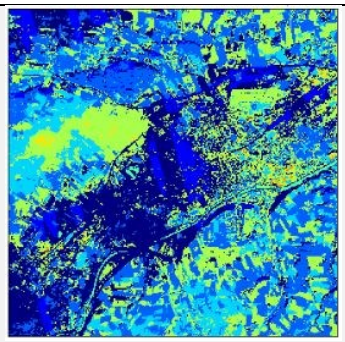
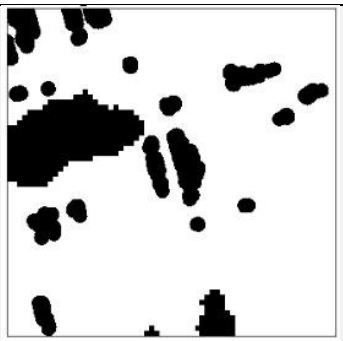
Water Vapor-, Scene Classification Maps and Quality Indicators for cloud and snow probabilities. In the context of this work SCL is used to detect valid data pixel. It contains the classes presented in table 3.

Table 3: SCL-Classes

Label	***Class***
1	SC_SATURATED_DEFECTIVE
2	SC_DARK_FEATURE_SHADOW
3	SC_CLOUD_SHADOW
4	VEGETATION
5	NOT-VEGETATED
6	WATER
7	SC_CLOUD_LOW_PROBABILITY / UNCLASSIFIED
8	SC_CLOUD_MEDIUM_PROBABILITY
9	CLOUD_HIGH_PROBABILITY
10	THIN_CIRRUS
11	SNOW

Cloud information is also added for the detection of invalid pixels. Cloud detection is performed at 160 m resolution. The cloud probability map and the corresponding cloud binary mask (CLM) are scaled to EOPatch's resolution. The eo-learn library is utilising a certain classifier for computing the respective cloud probability maps. The provided classifier in this case is the pixel-based S2 cloud detector "S2PixelCloudDetector" which takes Sentinel-2 images of all requested bands as input and returns a raster binary cloud mask, where 0 (1) indicates clear sky (cloudy) pixel. To filter valid reflectance data of each timeframe, an equally shaped mask is necessary. First, a mask is created based on the CLM. Second, valid classes are defined in the SCL (classes: 2, 4, 5, 6, 7) and on this basis another valid data mask is created. Finally, the two masks are combined to a complete mask. For illustration CLM, SCL and the final combined valid data mask are displayed in table 4.

Table 4: Cloud Masking

CLM	SCL	Final valid data mask
		

Based on the final mask created for each timeframe, frames containing too many invalid pixels are excluded from further processing. In this case all frames with a valid coverage of 70% will be kept. Figure 11 displays the number of timeframes per EOPatch in more detail and the average can be deduced. It is 25 of 54 possible frames in the corresponding time period.

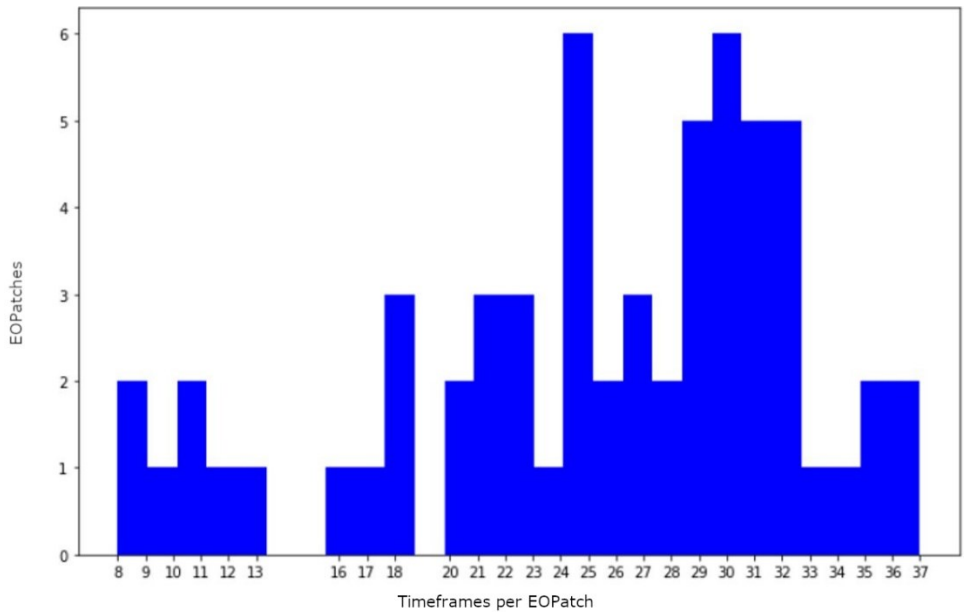


Figure 11: Timeframes per EOPatch

This is mainly caused by problematic cloud cover. Also, the recordings are not distributed regularly. Especially in the early months of the year there are drastic bad weather influences. These limitations will become relevant in later processing steps.

2.2.3 Reference Data

The LPIS reference data for 2018 is downloaded in vector format from the Geopedia (Geopedia homepage, https://geopedia.world/) via Sentinel-Hub tasks applied in the Perceptive Sentinel library. It is added to the EOPatch automatically. For a detailed investigation of the dataset "InVekoS Schläge Österreich 2018" can be downloaded manually (InVekoS dataset, https://www.data.gv.at/katalog/dataset/f7691988-e57c-4ee9-bbd0-e361d3811641). Figure 12 shows an EOPatch specific section of the reference data. With the help of the eo-learn library, the dataset is converted into the raster format. The raster dimension is assigned to the shape of the spectral raster as the actual purpose of the process is to label the respective crop types on a pixel basis.

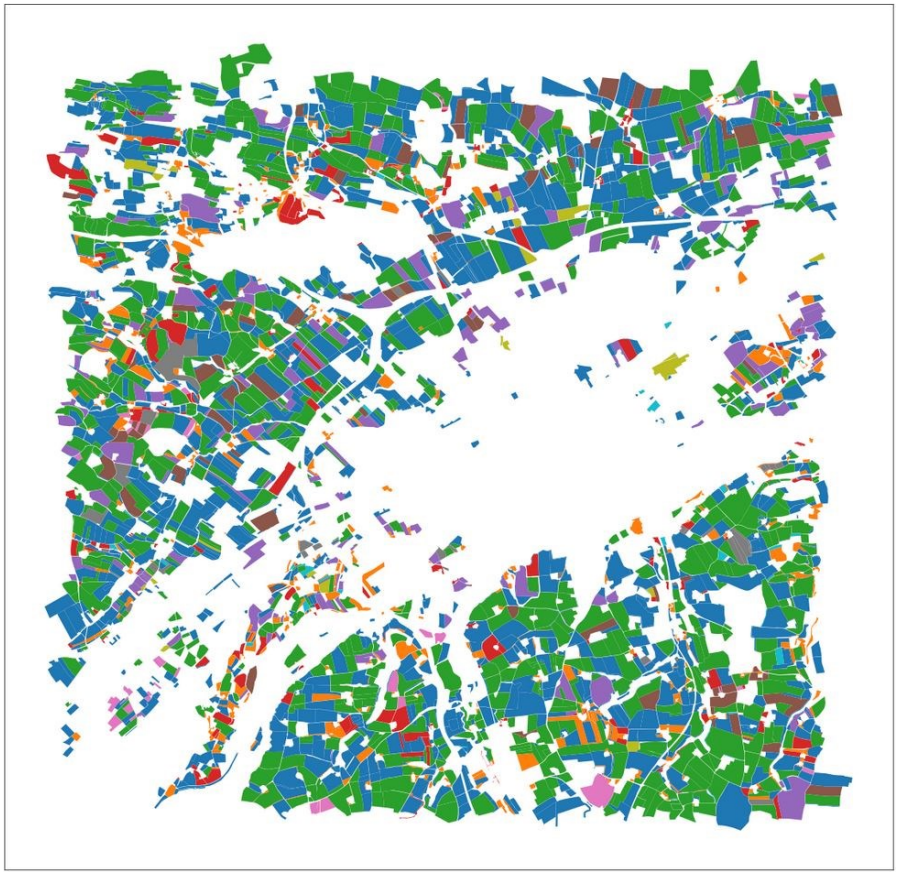


Figure 12: LPIS vector data

The over 200 LPIS classes are grouped according to chapter 1.3.1 which describes the respective grouping process. As it is the aim of this work, amongst other research questions, to achieve the most accurate and meaningful classification results the over 200 LPIS classes need to be further categorized. The initial LPIS classes stand for the main crops cultivated in the growing season. Potential intermediate crops are ignored for the classification. In the appendix of this thesis there is table 5 containing all LPIS classes as well as an overview of the grouping.

With completion of the categorization a buffer is applied to each individual field in the border area. This erosion is carried out to the extent of one pixel, i.e. 10 m. The necessity of this step is a result of the mixed pixel problem. Especially, in the border area of fields there are often green strips, hedges, or a direct transition to the neighbouring field. Accordingly, a heterogeneous spectral reflection occurs in this area. A pixel can contain both corn and grass, for example, thus forming a mixed pixel. To exclude these confusing pixel values from further processing, they are completely removed. After the LPIS vector data has been read and processed, it is stored in the EOPatch.

2.2.4 Feature Engineering

At least for the reference classification some feature engineering is necessary. First, from all available bands three different indices are calculated to potentially improve the results. Common DT based algorithms like LightGBM only approximate interactions and non-linear relationships between different bands. Only a binary split is performed on a single covariate at each step, all splits are orthogonal and therefore more complex interactions among covariates are less probable to be considered. Strong relationships benefit from being explicitly defined (Inglada et al. 2017). Indices therefore help to improve the exploitation of those (Pelletier et al. 2019). CNN should principally be able to identify all spectral relationships independently. These considerations are part of the analyses in chapter 1.4. Two of the three indices represent Normalised Difference Indices that are also referred to as Vegetation Indices. These are

calculated from two bands A and B, with the formula $\frac{(A-B)}{(A+B)}$. The first one, and most widely used in terms of vegetation classification, is the NDVI. It quantifies the characterizing "value jump" between green leaf scattering in the Near Infra-red (NIR) wavelength (S2-band: B08) and chlorophyll absorption in the visible (VIS) red wavelength (S2-band: B04). Strong and well-nourished vegetation will absorb most of the visible wavelengths it receives and will reflect a large proportion of the near-infra-red light, whereas vegetation in poor condition or sparsely vegetated areas, will reflect more radiation from the visible spectrum and less from near-infra-red.

The second vegetation index is the NDWI. It is typically used for water body mapping. The index uses the green (VIS) (S2-band: B03) and NIR (S2-band: B08) bands of remote sensing images based on this phenomenon. However, it is also suitable as a benchmark for the health of a plant (European Commission) as the test is based on the fact that senescing vegetation is highly reflective in NIR and more highly reflective in the green than green vegetation due to chlorophyll loss (ESA Technical Guides). Although it only detects this on the surface, since none of the target categories contains forest, this is not a problem in the present case. On the contrary, it has already been shown in studies to be superior to NDVI (Jackson 2004). Disadvantages can rather be caused by soil background effects related to coarse plant cover.

The third index is the NORM. The index normalizes the euclidean distance between spectral signatures of image pixels. The euclidean distance is zero when signatures are identical and tends to increase according to the spectral distance of signatures (Congedo). All indices are ultimately stored in the EOPatch.

2.2.5 Interpolation

After downloading the satellite data, calculating the indices, and adding all data to the EOPatch, it now contains 13 features. To receive useful results for the reference classification using the

LightGBM algorithm, the data within the timeframes of all EOPatches must be available in equidistant time intervals. So far, each EOPatch has an individual number of timeframes due to the individual valid data masks (Figure 11). To align the single datasets, a linear interpolation is performed. Initially it is determined which recording date over the year is the earliest or latest date that is present in all EOPatches. For the available data, these key dates take on the values 13.04.2018 and 20.09.2018. Within this period, equidistant timeframes are created for each EOPatch. This is done based on the respectively existing timeframes. During this work, an 8-day equidistant range is resampled. If there is not a recording for every eighth day, these missing values are interpolated. This results in a timeframe count of 21 for all EOPatches. Finally, the data is sampled, combined, and transformed into a format suitable for the respective algorithm in the following steps.

2.2.6 Sampling

To generate a stable and well generalizing model, it is necessary to randomly select individual pixels from the complete dataset and combine them into a sampling dataset. The first step is to determine which classes are significantly represented in the respective areas. All classes that are represented with at least 1000 pixels per EOPatch are included in the respective sampling process. The final number of samples (Figure 13) depends on how many classes are included and how many samples per class should be sampled. For example, if 20 classes are significantly represented and 500 pixels are to be sampled in each case, the resulting data set will have a size of 4000 samples. The eo-learn library can be used for sample selection. It selects random pixels and adds them to the dataset. Finally, the EOPatches must be assigned to the training or test dataset. The classification is done at EOPatch level to ensure that training and test samples do not come from the same field, which increases the overall confidence in the classification results (Kamilaris and Prenafeta-Boldú 2018). In this case the dataset is divided into 3/4 training data and 1/4 test data. In other words, every fourth EOPatch is used for the test data set

(Figure 1 – red/ blue squares). Finally, it is ensured that only those classes are included in the classification process that are present in both the training and the test dataset.

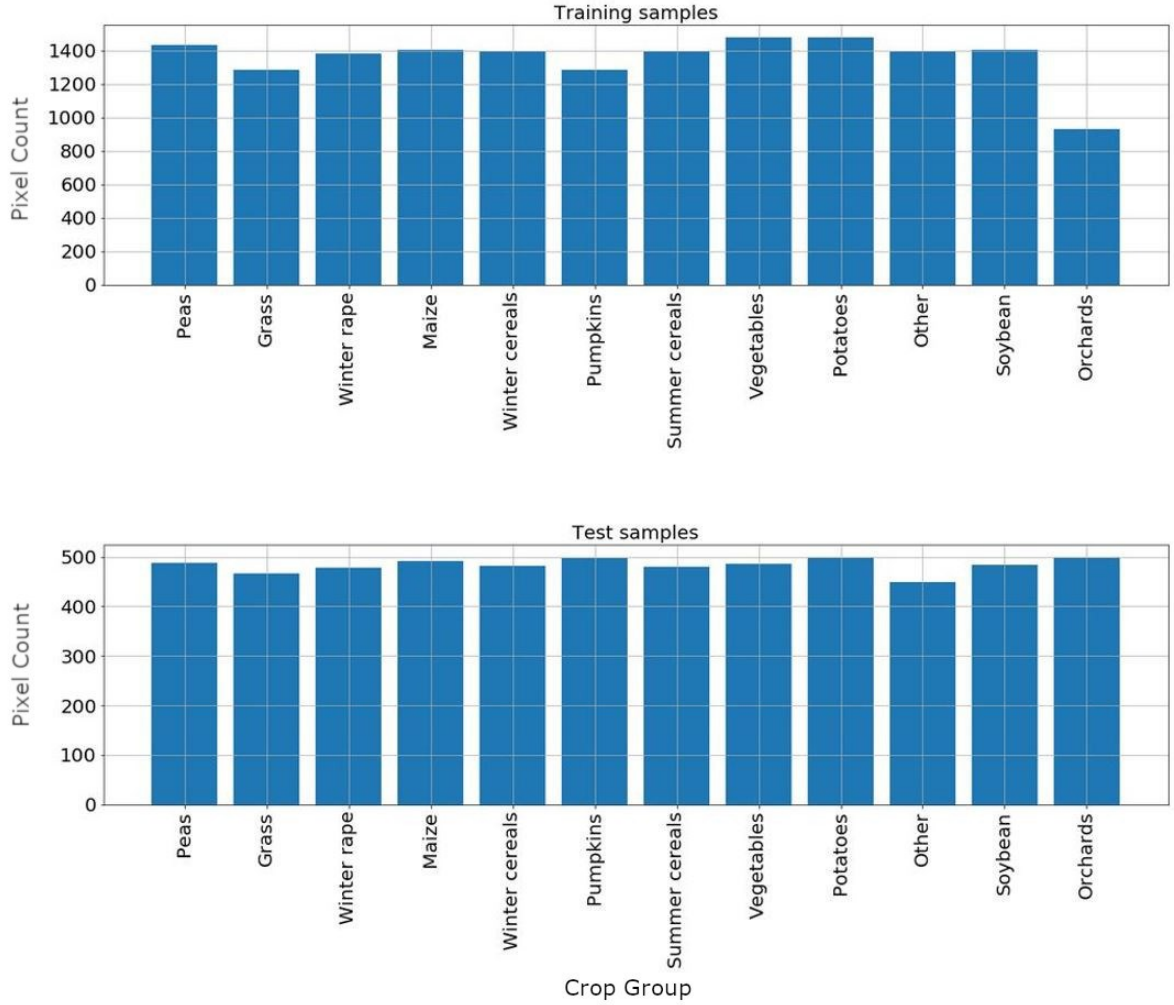


Figure 13: Classes distribution in pixel per class for training and test dataset

The two datasets now contain all necessary data and can be fed into the LightGBM algorithm in the next step.

2.3 Classification

Artificial Intelligence, Machine Learning and Deep Learning are just a few of the terms which regularly cause confusion among non-experts. This is not surprising as they share common ground in terms of software development. In conventional programming a practitioner gives specific instructions to a machine to find a solution for a given problem. In the case of the

mentioned terms, the machine is not getting any instructions for solving problems but is instructed to learn and find the best of all possible solutions. As illustrated in figure 14, AI is the general term for different approaches to simulate human behaviour or at least cognitive abilities by machines. In addition to fields like robotics or linguistics, this term also includes data analysis using machine learning. Thus, it can be said that all machine learning counts as AI, but not every AI application as machine learning. According to the requirements of this thesis only classification algorithms are illustrated. Depending on the respective task and dataset, approaches like Support-Vector-Machine (SVM) (Cortes and Vapnik 1995), K-Nearest Neighbor (k-NN) (Fix and Hodges 1951) or Decision Tree algorithms (DT) (Gordon et al. 1984) may be the methods of choice. In the context of this thesis an established decision tree-based algorithm is applied. The LightGBM is a Gradient Boosting Machine, which is a variant derived from the RF (Breiman 2001). Further, an approach from the field of deep learning is applied. Deep learning, like the decision tree-based algorithms, forms a subset of machine learning. Often this branch is reduced to deep artificial neural networks. Here, "deep" is mainly a technical term. It is derived from the "depth", i.e. the multiple count of layers of a neural network. Under this term, however, numerous other algorithms exist such as Recurrent Neural Networks (RNN) (Rumelhart et al. 1986) and Long Short-term Memory algorithms (LSTM) (Hochreiter and Schmidhuber 1997). In the following a variant of a Convolutional Neural Network (CNN) architecture will be used for a comparison with the LightGBM.

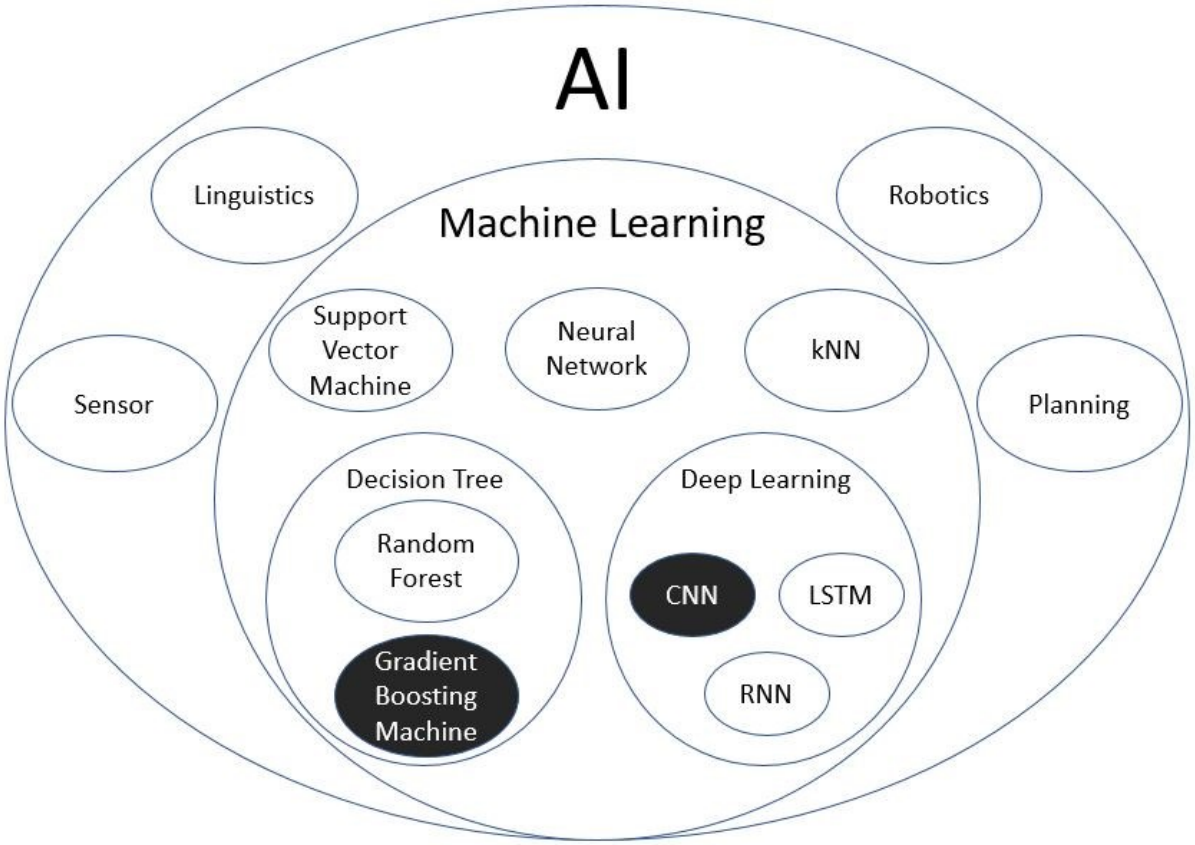


Figure 14: Overview AI

2.3.1 Reference Classification

The reference classification provides the basis for this thesis in terms of data pre-processing and accuracy assessment. As already stated, the applied LightGBM algorithm represents a state-of-the-art approach in the field of supervised classification tasks, which reliably and effectively leads to satisfactory results. To explain the exact functionality of this high-performant algorithm in an accessible way, it is useful to start with the fundamental element, a simple DT. Its basic flowchart-like tree structure, displayed in figure 15, is divided into three parts: internal decision nodes representing features or attributes, branches representing decision rules and leaf nodes which all together represent the outcome. It is possible for this tree to first select the attribute that is best suited to subdivide the record. This attribute then becomes a decision node and the data record is divided into smaller subsets. A tree is created from these processes by repeating

the process until either all tuples belong to the same attribute value, there are no more attributes or there are no more instances. DT are very intuitive but individually they are prone to overfitting and do not generalize well.

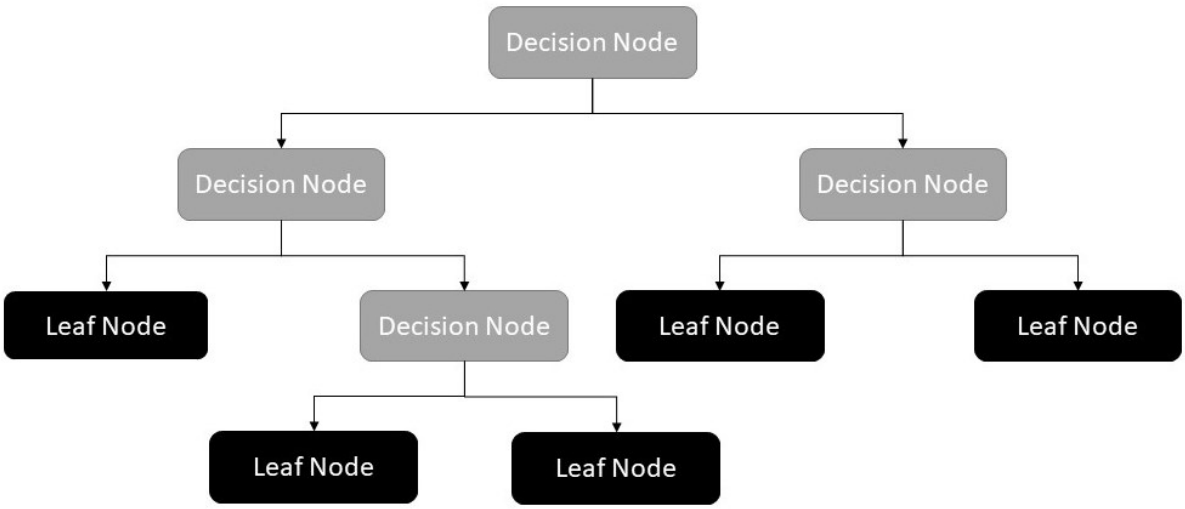


Figure 15: Decision Tree

Ensemble classifiers are preferred in remote sensing to compensate the shortcomings of individual DT. The idea is to learn multiple weak classifiers to generate one with a strong decision rule. One of the most popular ensemble classifiers for LULC classification is the RF. In the initial approach by Breiman (1996) many individual DTs operate as an ensemble. The underlying concept here is called Bootstrap-Aggregation or "bagging". When bagging is applied, the trees are generated parallelly on a subset of training samples through replacement. Each DT individually predicts a class and the mean of all predictions inside the architecture forms the result. The concept is simple but powerful: low correlation between the trees assures protection from individual errors and enables more accurate results. Besides, bagging, "boosting" is one of the most popular ensemble methods. As already stated in chapter 1.2.3.1 approaches such as Breimanns earlier introduced RF are currently often outperformed by state-of-the-art GBDT methods. Instead of parallel training (bagging) the models (DTs) are sequentially trained and therefore able to avoid errors made by the previous models. While

AdaBoost is updating the weight of wrongly classified points to learn, LGBM uses the residual error directly. The process starts by training a DT. Thereafter the tree is applied to predict, calculate the residual error, and use it as input data for the next prediction. These steps are repeated until the set number of trees is reached. For the final prediction Gradient Boosting adds up the predictions of all trees (Xia et al. 2017).

The LightGBM algorithm is an implementation of GBDT algorithm released by Microsoft. Most DT-based algorithms grow trees level- or depth-wise (Figure 16) to maintain a balanced tree. LightGBM in contrary grows trees leaf-wise or best-first (Figure 17) to maximize loss reduction. Another adaption concerns the calculation of the gain for each split in an internal node. Split means the decision for a feature or attribute based on maximum information gain. To find exactly the best of all possible decisions, in most architectures' algorithms such as, most popular, "Pre-sorted algorithm" are applied. It enumerates all possible decisions on pre-sorted values. This method is simple but computationally inefficient. LightGBM approximates the best decision. Therefore, it uses the histogram-based algorithm, which buckets continuous features into discrete bins for constructing feature histograms during training. What differs most from its closest relative, XGBoost, is its ability to subsample the data. When it comes to the split, the Gradient-Based One-Side Sampling (GOSS) allows instances with large gradients (i.e. under-trained instances) to be preferred to those with small gradients (close to local minima). Thereby, preference is given to instances, which increase information gain. In addition, LightGBM uses Exclusive Feature Bundling (EFB). The algorithm identifies features, which never take zero values simultaneously, bundles them into a single feature and thereby reduces the training complexity (Guolin Ke et al. 2017).

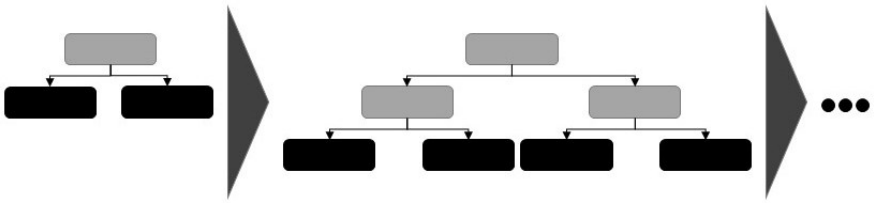


Figure 16: Level-wise growth

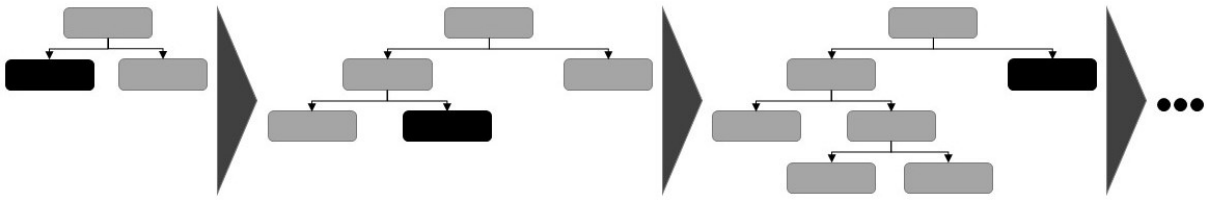


Figure 17: Leaf-wise growth

The architecture and its parameters applied in the context of this thesis are oriented towards the basic structure provided by scikit-learns' Python API for the LightGBM (LightGBM repository, https://github.com/microsoft/LightGBM). The codesubset 1 demonstrates the implementation of the LightGBM. $len(rev\_y\_train\_unique)$ indicates the number of classes considered for classification. X_train_lgbm includes all features and rev_y_train includes the respective class labels from the reference dataset. Except for the $objective$ and the $metric$ of multiple classes the default parameters are applied.

CodeSubset 1: LightGBM – Setup and training

```
# Set up the LightGBM model
model_lgbm = lgb.LGBMClassifier(
    objective='multiclass',
    num_class=len(rev_y_train_unique),
    metric='multi_logloss'
)

# Train the model
model_lgbm.fit(X_train_lgbm, rev_y_train)
```

2.3.2 TempCNN Classification

Considering the ideas behind neural networks in general, the main intention behind it is an adaption of the human brain. Instead of biology on a miniature scale, a set of algorithms is designed to recognize patterns. A basic Artificial Neural Network (ANN) architecture is shown in figure 18 as an example of a fully connected neural network. "Fully-connected" means, each layer takes all output of the previous layer as input. The first input layer represented by the green neurons contains the instances of the real-world data. Its size depends on the number of instances. The output layer or SoftMax represented by the red neurons has several units equal to the number of classes used in the classification task. In between there are two "hidden layers" represented by blue neurons. These layers, in number and size, are defined by the applicant.

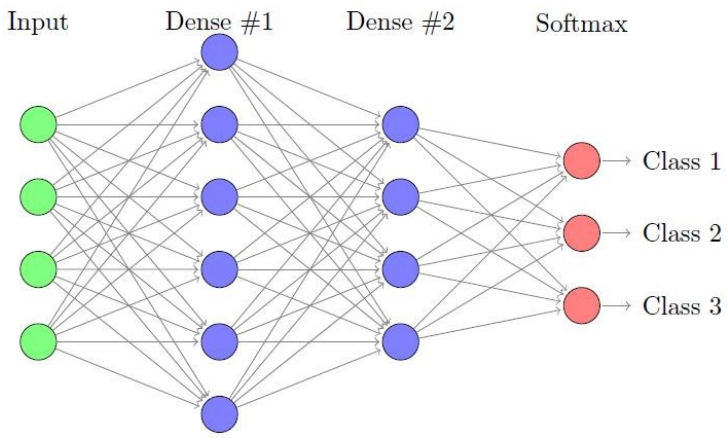


Figure 18: Artificial Neural Network

The real-world patterns are translated into numerical data stored in vectors. A vector is fed into the multi-layer network. Each layer has multiple units referred to as "neurons", which are "activated" in a certain order depending on the patterns in the input vector. This is also where the "learning" of the network takes place. Before the input vector is connected to the following neurons, it is calculated against a vector of "learning parameters". This vector is transformed frequently depending on the accuracy of the network's prediction. Expressed formally there are several layers l each containing an activation map $A^{[l]}$. As already mentioned, for the first layer

$A^{[0]}$ corresponds to the instances of the real-world input data. In the following layers $A^{[l]}$ refer to the activation map vector $A^{[l-1]}$ being multiplied with weights $W^{[l]}$, added a bias $b^{[l]}$ and applied to a non-linear activation function $g^{[l]}$.

$$A^{[l]} = g^{[l]}(W^{[l]}A^{[l-1]} + b^{[l]})$$

Using a non-linear activation function is crucial, as it allows to efficiently stack multiple layers. The exclusive use of linear functions would only result in a linear combination of the input as final output no matter the depth of the model. Stacking several layers keeps each single layer simple (i.e. small number of units) while increasing the capacity of the network for representing complex functions. In compliance with the work of Pelletier et al. (2019) the Rectified Linear Units function (ReLU) is used.

The final goal of a neural network is to minimize a given cost function J. J compares the actual labels of the data y_i to the predicted ones $\acute{y}_i$, evaluates the committed erros and thereby assesses the fit of the model to the data. For minimizing J optimal values for weights $W^{[l]}$ and $b^{[l]}$ need to be found.

$$\mathrm{J(W, b)} = \frac{1}{n}\sum_{x_i} \mathcal{L}(\acute{y}_{i,}\, y_i)$$

Cross-entropy loss is used as loss function $\mathcal{L}$. Most often it is the best choice for multi-class classification tasks and therefore represents the most popular one. Formally it looks as follows:

$$\mathcal{L}(\acute{y}_i, y_i) = -\log\,(p(y_i|x_i))$$

where $p(y_i|x_i)$ represents the probability of predicting the true class y_i of the instance i from the softmax of the network.

It should be remembered that it is at this point that DL is constituted. An ANN consisting of three layers (1 input-, 1 hidden-, 1 output-layer) is still accounted as one of the general ML

approaches (Figure 14). Only with the implementation of multiple hidden (i.e. convolutional or pooling) layers, the "depth" of a model is generated which constitutes the concept of DL. This "depth" is crucial for the functional advantages of DL. An ANN with more than two hidden layers provides the required complexity to learn its own feature representations from the input data within a discriminative end-to-end model. It is therefore not dependent on manual feature engineering or preceding models (LeCun and Bengio 1998). Speaking of the models self-generated features, in CNN architectures like in figure 19 convolutional layers are preceding the neural network.

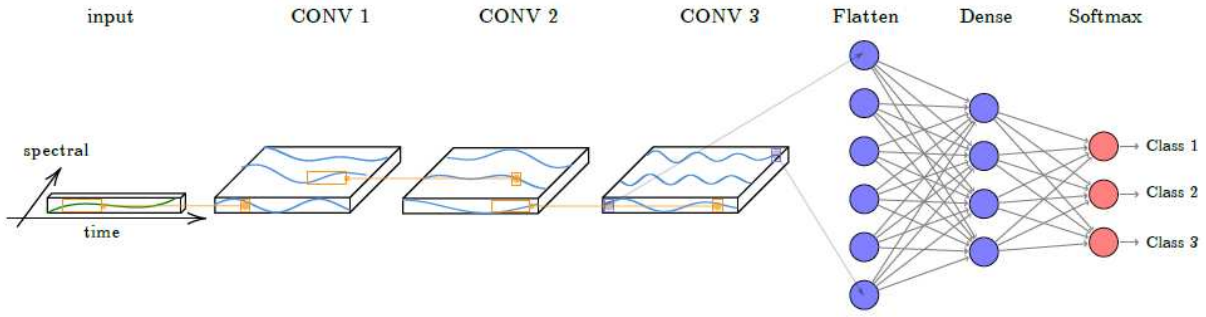


Figure 19: Temporal Convolutional Neural Network architecture

Their task is to limit the parameters that the network must learn. By applying a convolution filter on the output of the previous layer, an activation map (not to be confused with the activation value in hidden layers) is created as output matrix. In the context of their original purpose of object recognition in images, their task is to filter out certain features, such as vertical and horizontal edges, from the pixel values. Decisive for the dimension of the respective output matrix of a filter are both, the output matrix (image) and the filter matrix. In addition, the way the filter runs over the image is relevant, i.e. whether "padding" (extension of the output matrix) or a "stride" (adjustment of the number of pixels that are skipped by the filter in the output matrix) is applied. Filtering a 3D matrix, e.g. an RGB image, works with the help of a 3D filter. Each channel of the output image can be examined with different filter methods (vertical, horizontal, etc.). The result is 1-dimensional for each filter. The more different features are

examined the more different filters are applied and the higher the dimension of the result matrix gets. A further type of layer potentially preceding the neural network, next to the convolutions, is the pooling layer. This layer combines pixel groups. Here the hyperparameter filter size, stride, and the choice between Max- and Average Pooling are decisive. Average pooling is mainly used in deep CNN to reduce high dimensions, while max pooling highlights edges.

The previously described characteristics of CNN brought significant success in image recognition. However, these classical structures have limited suitability for the exploitation of SITS. To address this deficit, the idea of one-dimensional convolutions will be adapted in this work. Figure 20 shows how time series information is converted from a filter (in this case a "Gradient Extraction Filter" – [-1 -1 0 1 1]) to model-specific features. The complex SITS is converted to a simpler feature.

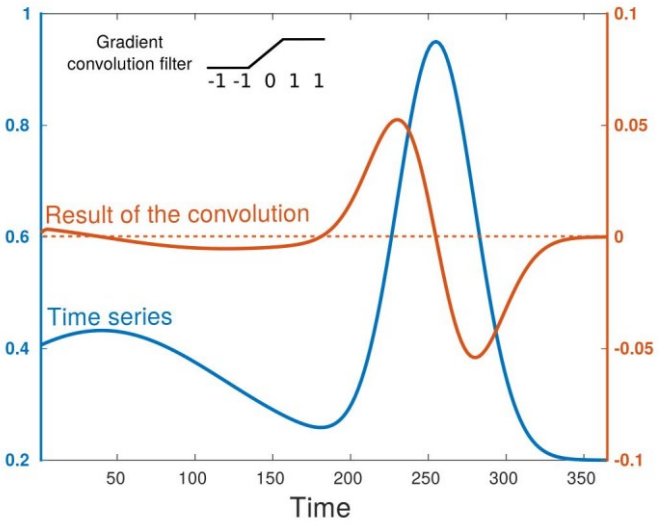


Figure 20: Convolution result of Gradient Extraction Filter [-1 -1 0 1 1]

Figure 21 illustrates how different convolutional filters run across the input matrix. The initial filter matrix is represented as T=length of SITS x D=number of features]. In the case of this work the matrix would look like this: (T=21 x D=13). The first illustration, "No guidance", represents similarities between CNN and conventional algorithms such as RF or ANN. Neither spectral nor temporal structures are recognized in the data set.

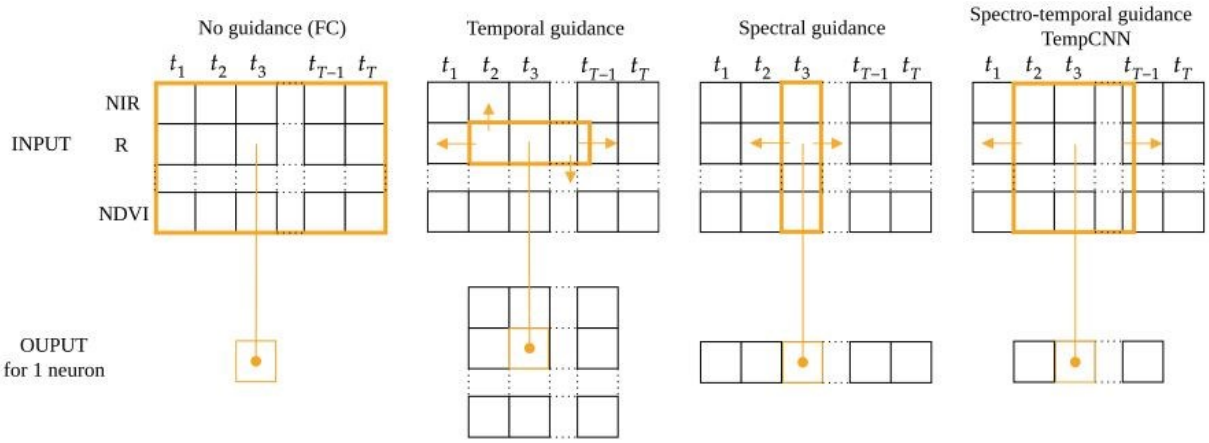


Figure 21: Convolutional filter guidance

With the second graphic of "Temporal guidance" a link to the previous figure 20 can be made. As the filter is [-1 -1 0 1 1] it covers a period of 5 timeframes which results in the dimensions [T=5 x D=1]. For all spectral features, the same filter is applied over the time dimension. The "Spectral guidance" illustration shows the application of a filter with the dimensions [T=1 x D=number of features], considering the spectral characteristics of one timeframe. The most relevant representation for this work is "Spectro-temporal guidance". As the name already suggests, spectral and temporal patterns are detected. If tf is an optional number of timeframes the dimensions can be described as [T=tf x D=number of features].

Engineering the architecture of a network like this, choosing its hyperparameters (number of convolutional-, pooling-, and hidden-layers, dropout, learning rate, etc.) and deciding how to optimize it requires significant expertise. Considering that challenge, practically complexity is shifted from feature to architecture engineering. As there were multiple architectures published in the recent past, the most efficient way of running a DL model is to apply an existing architecture. As already mentioned in this thesis the architecture and parameters engineered and evaluated by Pelletier et al. (2019) are applied. The codesubset 2 demonstrates the implementation of the TempCNN. X_train and y_train include the training features and reference labels. X_test and y_test include validation data, respectively. Apart from the

number of output filters of 5, a kernel size of 3, the ReLU activation function for all layers except the SoftMax layer, a dropout rate of 0,5, the categorical cross entropy loss function and Adam optimization the default architecture values were adopted. For training the number of epochs is set to 10, the batch size is set to 32 and the training samples are shuffled.

CodeSubset 2: TempCNN – Setup and training

```
# Set up the TempCNN architecture
model_tcnn = Sequential()
model_tcnn.add(Conv1D(filters=5, kernel_size=3, activation='relu',
input_shape=(n_timesteps,n_features)))
model_tcnn.add(Dropout(0.5))
model_tcnn.add(Conv1D(filters=5, kernel_size=3, activation='relu',
input_shape=(n_timesteps,n_features)))
model_tcnn.add(Dropout(0.5))
model_tcnn.add(Conv1D(filters=5, kernel_size=3, activation='relu',
input_shape=(n_timesteps,n_features)))
model_tcnn.add(Dropout(0.5))
model_tcnn.add(Flatten())
model_tcnn.add(Dense(256, activation='relu'))
model_tcnn.add(Dense(label_count, activation='softmax'))
model_tcnn.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])

# Train the model
model_tcnn.fit(X_train,
          y_train,
          validation_data=(X_test, y_test),
          epochs=10,
          batch_size=32,
          verbose=1,
          shuffle=True)
```

2.4 Accuracy Assessment

The central aim of this thesis is the evaluation of TempCNN and the comparison to a state-of-the-art ML algorithm. In the context of a ML classification task an extensive accuracy assessment is crucial for quantifying experimental findings and thus evaluating the performance of the classifiers. For the elaboration of the research questions specified in chapter 1.1.2, four assessments are performed. The initial step involves the crop grouping. Within the Perceptive Sentinel project, the participating partners initially agreed on a preferable categorization. This agreement intends a categorization into 25 individual groups. On this basis, the further grouping

process aims at identifying context-based groups that can also be spectrally delimited. For this purpose, the LPIS crop classes are recombined several times until both, context and overall accuracy, are adequate. This procedure results in the 5 steps outlined in chapter 1.3.1.

The context is also of major importance in the accuracy assessment considering the heterogenous crop groups described in chapter 1.3.2. The three categories of heterogeneity are not based on quantifiable characteristics of the individual crop classes, but on meta-information about cultivation practices, spectral inconsistency, and intersection.

Assessing the performance of the classifiers against the background of feature engineering, one reference is expected in the temporal resolution of the SITS. The respective results are evaluated in chapter 1.3.4. An 8-day interval is chosen as a standard for the general processing workflow as it enables processable amounts of data at a high temporal resolution. To identify potential performance differences at higher resolution, a 4-day interval is applied. This adjustment allows a still reasonable data volume combined with more detailed SITS. A higher resolution does not increase the accuracy significantly, because of the Sentinel-2 revisit frequency of up to 5 days and unnecessarily consumes processing capacity. An interval greater than 8-day is not an option either, as it results in an unreasonable loss of temporal information, with a small gain in processing capacity.

The idea behind the assessment of the classifier's performance facing presence and absence of spectral features is simple. As already stated in chapter 2.2.4, Pelletier et al. (2019) expect their TempCNN architecture to independently identify all relationships between the different spectral bands. In this case, a basic test with presence or absence of precalculated indices provides valuable results evaluated in chapter 1.3.3.

To assess the spatial stability of the models derived by the two classifiers, data from outside the training area is required. Determining the extent and spatial distribution of the data is a decisive

first step. In the context of this work 43 EOPatches (Figure 1 – yellow squares) are manually selected from agricultural areas distributed throughout Austria. Manual selection is required because of the specific agricultural conditions in the alpine region. Narrow valleys make it more difficult to find a square area of 100 km², which is more than 30% cultivated. Therefore, the majority of EOPatches is located in the plain regions between the training areas. Since this last assessment is rather explorative in character, the evaluation is visually qualitative, unlike the previous quantitative steps. The performance comparison per crop group and EOPatch is considered in relation to the distribution of the individual LPIS classes within the respective crop group (Figure 9 and 10). On this cartographic basis, visually recognizable patterns can be identified.

References

Breiman, L. (2001): Random Forests. In: *Machine Learning* 45 (1), S. 5–32. DOI: 10.1023/a:1010933404324.

Congedo, Luca: Introduction to Remote Sensing. URL: https://semiautomaticclassificationmanual-v4.readthedocs.io/en/latest/remote_sensing.html#classification-algorithms, Last accessed: 27.08.2020.

Cortes, Corinna; Vapnik, Vladimir (1995): Support-vector networks. In: *Machine Learning* 20 (3), S. 273–297. DOI: 10.1023/a:1022627411411.

ESA Plugins: Sen2Cor Plugin. URL https://step.esa.int/main/third-party-plugins-2/sen2cor/, Last accessed: 27.08.2020.

ESA Technical Guides: L2A algorithm. URL: https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm, Last accessed: 27.08.2020.

European Commission: Factsheet NDWI. URL: https://edo.jrc.ec.europa.eu/documents/factsheets/factsheet_ndwi.pdf, Last accessed: 27.08.2020.

Fix, Evelyn; Hodges, J. L. (1951): Discriminatory analysis: Nonparametric discrimination: Consistency properties.

Gordon, A. D.; Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984): Classification and Regression Trees. In: *Biometrics* 40 (3), S. 874-889. DOI: 10.2307/2530946.

Guolin Ke; Qi Meng; Thomas Finley; Taifeng Wang; Wei Chen; Weidong Ma et al. (2017): LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: *Advances in neural information processing systems*, S. 3146–3154.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. In: *Neural computation* 9 (8), S. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Inglada, Jordi; Vincent, Arthur; Arias, Marcela; Tardy, Benjamin; Morin, David; Rodes, Isabel (2017): Operational High Resolution Land Cover Map Production at the Country Scale Using Satellite Image Time Series. In: *Remote Sensing* 9 (1), S. 95-116. DOI: 10.3390/rs9010095.

Jackson, T. (2004): Vegetation water content mapping using Landsat data derived normalized difference water index for corn and soybeans. In: *Remote Sensing of Environment* 92 (4), S. 475–482. DOI: 10.1016/j.rse.2003.10.021.

Kamilaris, Andreas; Prenafeta-Boldú, Francesc X. (2018): Deep learning in agriculture: A survey. In: *Computers and Electronics in Agriculture* 147, S. 70–90. DOI: 10.1016/j.compag.2018.02.016.

LeCun, Yann; Bengio, Yoshua (1998): Convolutional networks for images, speech,and time-series. In: The Handbook of Brain Theory and Neural Networks, S. 255–258.

Maponya, Mmamokoma Grace; van Niekerk, Adriaan; Mashimbye, Zama Eric (2020): Pre-harvest classification of crop types using a Sentinel-2 time-series and machine learning. In: *Computers and Electronics in Agriculture* 169, S. 105–164. DOI: 10.1016/j.compag.2019.105164.

Pelletier, Charlotte; Webb, Geoffrey; Petitjean, François (2019): Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series. In: *Remote Sensing* 11 (5), S. 523-548. DOI: 10.3390/rs11050523.

Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (1986): Learning representations by back-propagating errors. In: *nature* 323 (6088), S. 533–536. DOI: 10.1038/323533a0.

Xia, Yufei; Liu, Chuanzhe; Li, YuYing; Liu, Nana (2017): A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. In: *Expert Systems with Applications* 78, S. 225–241. DOI: 10.1016/j.eswa.2017.02.017.

Appendix

Table 5: Crop grouping

LPIS Crop Class	***Perceptive Sentinel Crop Group***	***Final Crop Group***
Mountain mower	Alpine meadows	Grass
Fodder grasses *Grass roll* *Once per year mow meadow* *Crop rotation- natural vegetation without planted vegetation* *Crop rotation - non cultivated for some time* *Different green areas* *Changing meadow*	Grass	
Common sainfoin *Clover* *Grass clover mixture* *Alfalfa* *Lacy phacelia* *Flat peas*	Leafy Legumes and/or grass mixture	
3 or more times mowed meadows *2 times mowed meadows* *Natural meadow not for animal consumption*	Meadows	
Alpine pasture *Permanent pastures* *Pasture* *Other pasture*	Pasture	
Maize *Maize (fodder)* *Maize (fodder) / in vegetation production* *Seed maize* *Maize for silage* *Sorghum* *Sweet maize*	Maize	Maize
Other fruit *Maroni - chestnut* *Cherry* *Apricot* *Nectarine* *Peaches* *Plums* *Quince* *Shell fruits* *Apples* *Pears* *Sour Cherry* *Flea*	Orchards	Orchards
Peas *Peas* *Sweet peas*	Peas	Peas
Early potato *Potato as a fodder* *Seed potato* *Potato - industrial and human consumption* *Potato / human consumption* *Potato for industrial production*	Potatoes	Potatoes
Pumpkin for oil *Pumpkin*	Pumpkins	Pumpkins
Soybean	Soybean	Soybean
Over summering emmer wheat or single grain wheat *Millet* *Summer Spelt* *Summer barley* *Summer oat* *Summer durum wheat* *Summer cereals* *Summer rye* *Summer triticale* *Summer wheat*	Summer cereals	Summer cereals
Sunflower	Sunflower	Sunflower

Field vegetable - uniform production *Cucumber as open field production* *Field vegetable production - mixture for fresh consumption and processing* *Field vegetable production - mixture* *Field vegetable production without harvesting* *Field vegetable - uniform production for processing* *Field vegetable production - mixture for processing*	Vegetables	Vegetables
Production of vine planting material *Vineyard (establish)* *Different vineyard areas* *Wine*	Vineyards	Vineyards
Overwintering emmer wheat or single grain wheat *Fresh rye as a fodder* *Winter spelt* *Winter spelt /vegetable production* *Winter barley* *Winter oat* *Winter durum wheat* *Winter cereals* *Winter rye* *Winter triticale* *Winter wheat*	Winter cereals	Winter cereals
Winter rape	Winter rape	Winter rape
Mixture of broad bean and cereals *Mixture of broad bean and peas* *Amaranth* *Other permanent crops* *Bee breeding fallow land* *Narrow leaf or blue lupin* *Flower and ornamental plants* *Flower and ornamental plants in tunnels* *Flower and ornamental plants in greenhouse* *One-year nursery* *Ornamental grasses* *Grasses for energy production* *Wood energy plantations without Robinia pseudoacacia* *Wood energy plantations with Robinia pseudoacacia* *Mixture of peas and cereals* *Mixture of peas and cereals or buckwheat* *Mixture of peas and cereals in vegetable production on the field* *Strawberry* *First forestation* *Forestation* *Common flax for processing* *Forest tree nursery - forest genetic resources* *Vegetable production under tunnel* *Vegetable production in greenhouse* *Fennel* *Herbs* *Herbs production under the tunnel* *Herbs production in the greenhouse* *Ginkgo* *Ditch banch* *Undefined* *Stone slope* *Small standing water* *Fresh maize as fodder* *Hemp* *Medicinal plants* *Medicinal plants in the tunnel* *Elderberry* *Saint John's wort* *Canary seed* *Chickpea* *Camelina* *Lentil* *Woody plants on the field trees bushes* *Evergreen hedges woody plants near the cost* *banch stone* *Intercropping beans and maize* *Milk thistle* *Nurseries* *Nature conservation area* *Fruit production in the tunnel*	Other	Other

Fruit production in the greenhouse *Crop rotation in orchards or hop production* *Flax* *Flax/in vegetable production* *Radish* *Quinoa* *Mustard* *Summer Caraway* *Summer rapeseed* *Summer turnip* *Common Vetch* *Other arable land* *Other arable plants* *Area of production under different protection* *Different production in the plastic tunnels* *Different production in the greenhouse* *Oleaginous fruits* *Special areas* *Different fodder* *Sudan grass* *Sweet lupin* *Topinambur* *Measures for forest and environment preservation* *Green manure to rise nitrogen content in the soil* *Mixed sowing od common vetch and cereals* *Winter caraway* *Turnip Tops* *Winter vetch*		
Summer Poppy flower *Winter Poppy flower*	Poppy	
Raspberries blackberries blueberries…	Soft fruits	
Hop	Hop	
Set aside for 20 years	Fallow land	
Buckwheat	Buckwheat	
Fodder Beet *Root beet for seed production*	Beets	
Broad beans	Beans	
Broad bean / in vegetable production *Over summering emmer wheat or single grain wheat in vegetable production on the field* *Overwintering emmer wheat or single grain wheat in vegetable production on the field* *Strawberry - in open field production* *Early potato following by buckwheat* *Early potato in vegetable field production* *Early potato following by maize* *Fodder grasses in vegetable production in open field* *Fresh rye as a fodder/following millet* *Fresh rye as a fodder/following maize* *Fresh rye as a fodder/following sudan grass* *Millet / in vegetable production* *Clover / in vegetable production* *Grass clover mixture / in vegetable production* *Peas / in vegetable production* *Summer barley following buckwheat* *Summer barley / in vegetable production* *Summer oat / in vegetable production* *Summer oat / fodder beet* *Summer durum wheat following buckwheat* *Summer durum wheat / in vegetable production* *Summer cereals / in vegetable production* *Summer wheat / in vegetable production* *Potato / human consumption in vegetable production* *Winter barley following buckwheat* *Winter barley / vegetable production* *Winter durum wheat following buckwheat* *Winter durum wheat / in vegetable production* *Winter rye / in vegetable production* *Winter triticale / in vegetable production* *Winter millet* *Winter wheat following buckwheat* *Winter wheat / in vegetable production* *Sweet maize / in vegetable production*	Multi use	No data