Article

Multi-Temporal Land Cover Classification with Sequential Recurrent Encoders

Marc Rußwurm * and **Marco Körner**

Chair of Remote Sensing Technology, TUM Department of Civil, Geo and Environmental Engineering, Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany; marco.koerner@tum.de
* Correspondence: marc.russwurm@tum.de; Tel.: +49-172-81-70-121

Abstract: Earth observation (EO) sensors deliver data at daily or weekly intervals. Most land use and land cover classification (LULC) approaches, however, are designed for cloud-free and mono-temporal observations. The increasing temporal capabilities of today's sensors enable the use of temporal, along with spectral and spatial features.Domains such as speech recognition or neural machine translation, work with inherently temporal data and, today, achieve impressive results by using sequential encoder-decoder structures. Inspired by these sequence-to-sequence models, we adapt an encoder structure with convolutional recurrent layers in order to approximate a phenological model for vegetation classes based on a temporal sequence of Sentinel 2 (S2) images. In our experiments, we visualize internal activations over a sequence of cloudy and non-cloudy images and find several recurrent cells that reduce the input activity for cloudy observations. Hence, we assume that our network has learned cloud-filtering schemes solely from input data, which could alleviate the need for tedious cloud-filtering as a preprocessing step for many EO approaches. Moreover, using unfiltered temporal series of top-of-atmosphere (TOA) reflectance data, our experiments achieved state-of-the-art classification accuracies on a large number of crop classes with minimal preprocessing, compared to other classification approaches.

Keywords: deep learning; multi-temporal classification; land use and land cover classification; recurrent networks; sequence encoder; crop classification; sequence-to-sequence; Sentinel 2

---

1. Introduction

Land use and land cover classification (LULC) has been a central focus of Earth observation (EO) since the first air- and space-borne sensors began to provide data. For this purpose, optical sensors sample the spectral reflectivity of objects on the Earth's surface in a spatial grid at repeated intervals. Hence, LULC classes can be characterized by spectral, spatial and temporal features. Today, most classification tasks focus on spatial and spectral features [1], while utilizing the temporal domain had long proven challenging. This is mostly due to limitations on data availability, the cost of data acquisition, infrastructural challenges regarding data storage and processing and the complexity of model design and feature extraction over multiple time frames.

Some LULC classes, such as urban structures, are mostly invariant to temporal changes and, hence, are suitable for mono-temporal approaches. Others, predominantly vegetation-related classes, change their spectral reflectivity based on biochemical processes initiated by phenological events related to the type of vegetation and to environmental conditions. These vegetation-characteristic phenological transitions have been utilized for crop yield prediction and, to some extent, for classification [2,3]. However, to circumvent the previously-mentioned challenges, the dimensionality of spectral bands has often been compressed by calculating task-specific indices, such as the normalized difference vegetation index (NDVI), the normalized difference water index (NDMI) or the enhanced vegetation index (EVI).

Today, most of these temporal data limitations have been alleviated by technological advances. Reasonable spatial and temporal resolution data of multi-spectral Earth observation sensors are available at no cost. Moreover, new services inexpensively provide high temporal and spatial resolution imagery. The cost of data storage has decreased, and data transmission has become sufficiently fast to allow gathering and processing all available images over a large area and multiple years. Finally, new advances in machine learning, accompanied by GPU-accelerated hardware, have made it possible to learn complex functional relationships, solely from the data provided.

Since now data are available at high resolutions and processing is feasible, the temporal domain should be exploited for EO approaches. However, this exploitation requires suitable processing techniques utilizing all available temporal information at reasonable complexity. Other domains, such as machine translation [4], text summarization [5–7] or speech recognition [8,9], handle sequential data naturally. These domains have popularized sequence-to-sequence learning, which transforms a variable-length input sequence to an intermediate representation. This representation is then decoded to a variable-length output sequence. From this concept, we adopt the sequential encoder structure and extract characteristic temporal features from a sequence of Sentinel 2 (S2) images using a straightforward, two-layer network.

Thus, the main contributions of this work are:

(i)   the adaptation of sequence encoders from the field of sequence-to-sequence learning to Earth observation (EO),
(ii)   a visualization of internal gate activations on a sequence of satellite observations and,
(iii)   the application of crop classification over two seasons.

2. Related Work

As we aim to apply our network to vegetation classes, we first introduce common crop classification approaches, to which we will compare our results in Section 6. Then, we motivate data-driven learning models and cover the latest work on recurrent network structures in the EO domain.

Many remote sensing approaches have achieved adequate classification accuracies for multi-temporal crop data by using multiple preprocessing steps in order to improve feature separability. Common methods are atmospheric correction [10–14], calculation of vegetation indices [10–14] or the extraction of sophisticated phenological features [13]. Additionally, some approaches utilize expert knowledge, for instance, by introducing additional agro-meteorological data [10], by selecting suitable observation dates for the target crop-classes [14] or by determining rules for classification [11]. Pixel-based [10,13] and object-based [11,12,14] approaches have been proposed. Commonly, decision trees (DTs) [10,11,14] or random forests (RFs) [12,13] are used as classifiers, the rules of which are sometimes aided by additional expert knowledge [11].

These traditional approaches generally trade procedural complexity and the use of region-specific expert knowledge for good classification accuracies in the respective areas of interest (AOIs). However, these approaches are, in general, difficult to apply to other regions. Furthermore, the processing structure requires supervision to varying degrees (e.g., product selection, visual image inspection, parameter tuning), which impedes application at larger scales.

Today, we are experiencing a change in paradigm: away from the design of physically-interpretable, human-understandable models, which require task-specific expert knowledge, towards data-driven models, which are encoded in internal weight parameters and derived solely from observations. In that regard, hidden Markov models (HMMs) [15] and conditional random fields (CRFs) [16] have shown promising classification accuracies with multi-temporal data. However, the underlying Markov property limits long-term learning capabilities, as Markov-based approaches assume that the present state only depends on the current input and one previous state.

Deep learning methods have had major success in fields, such as target recognition and scene understanding [17], and are increasingly adopted by the remote sensing community. These methods

have proven particularly beneficial for modeling physical relationships that are complicated, cannot be generalized or are not well-understood [18]. Thus, deep learning is potentially well suited to approximate models of phenological changes, which depend on complex internal biochemical processes of which only the change of surface reflectivity can be observed by EO sensors. A purely data-driven approach might alleviate the need to manually design a functional model for this complex relationship. However, caution is required, as external and non class-relevant factors, such as seasonal weather or observation configurations, are potentially incorporated into the model, which might remain undetected if these factors constantly bias the dataset.

In remote sensing, convolutional networks have gained increasing popularity for mono-temporal observation tasks [19–22]. However, for sequential tasks, recurrent network architectures, which provide an iterative framework to process sequential information, are generally better suited. Recent approaches utilize recurrent architectures for change detection [23–25], identification of sea level anomalies [26] and land cover classification [27]. For long-term dependencies, Jia et al. [24] proposed a new cell architecture, which maintains two separate cell states for single- and multi-seasonal long-term dependencies. However, the calculation of an additional cell state requires more weights, which may prolong training and require more training samples.

In previous work, we have experimented with recurrent networks for crop classification [28] and achieved promising results. Based on this, we propose a network structure using convolutional recurrent layers and the aforementioned adaptation of a many-to-one classification scheme with sequence encoders.

3. Methodology

Section 3.1 incrementally introduces the concepts of artificial neural networks (ANNs), feed-forward networks (FNNs) and recurrent neural networks (RNNs) and illustrates the use of RNNs in sequence-to-sequence learning. We then describe the details of the proposed network structure in Section 3.3.

3.1. Network Architectures and Sequential Encoders

Artificial neural networks approximate a function $\hat{y} = f(x; W)$ of outputs $\hat{y}$ (e.g., class labels) from input data x given a large set of weights W. This approximation is commonly referred to as the inference phase. These networks are typically composed of multiple cascaded layers with hidden vectors h as intermediate layer outputs. Analogous to the biological neural cortex, single elements in these vectors are often referred to as neurons. The quality of the approximation $\hat{y}$ with respect to ground truth y is determined by the loss function $L(\hat{y}, y)$. Based on this function, gradients are back-propagated through the ANN and adjust network weights W at each training step.

Popular feed-forward networks often utilize convolutional or fully-connected layers at which the input data are propagated through the network once. This is realized by an affine transformation (fully-connected) $h = \sigma(Wx)$ or a convolution $h = \sigma(W * x)$ followed by an element-wise, non-linear transformation $\sigma : \mathbb{R} \mapsto \mathbb{R}$.

However, domains like translation [4], text summarization [5–7] or speech recognition [8,9] formulate input vectors naturally as a sequence of observations $x = \{x_0, \ldots, x_T\}$. In these domains, individual samples are generally less expressive, and the overall model performance is based largely on contextual information.

Sequential data are commonly processed with recurrent neural network (RNN) layers, in which the hidden layer output h_t is determined at time t by current input x_t in combination with the previous output h_{t-1}. In theory, the iterative update of h_t enables RNNs to simulate arbitrary procedures [29], since these networks are Turing complete [30]. The standard RNN variant performs the update step $h_t = \sigma(W\tilde{x})$ by an affine transformation of the concatenated vector $\tilde{x} = [x_t \| h_{t-1}]$ followed by a non-linearity σ. Consequently, the internal weight matrix is multiplied at each iteration step, which essentially raises it to a high power [31]. At gradient back-propagation, this iterative matrix

multiplication leads to vanishing and exploding gradients [32,33]. While exploding gradients can be avoided with gradient clipping, vanishing gradients impede the extraction of long-term feature relationships. This issue has been addressed by Hochreiter and Schmidhuber [34], who introduced additional gates and an internal state vector c_t in long short-term memory (LSTM) cells to control the gradient propagation through time and to enable long-term learning, respectively. Analogous to standard RNNs, the output gate o_t balances the influence of the previous cell output h_{t-1} and the current input x_t. At LSTMs, the cell output h_t is further augmented by an internal state vector c_t, which is designed to contain long-term information. To avoid the aforementioned vanishing gradients, reading and writing to the cell state is controlled by three additional gates. The forget gate f_t decreases previously-stored information by element-wise multiplication $c_{t-1} \odot f_t$. New information is added by the product of input gate i_t and modulation gate j_t. Illustrations of the internal calculation can be seen in Figure 1, and the mathematical relations are shown in Table 1. Besides LSTMs, gated recurrent units (GRUs) [35] have gained increasing popularity, as these cells achieve similar accuracies to LSTMs with fewer trainable parameters. Instead of separate vectors for long- and short-term memory, GRUs formulate a single, but more sophisticated, output vector.
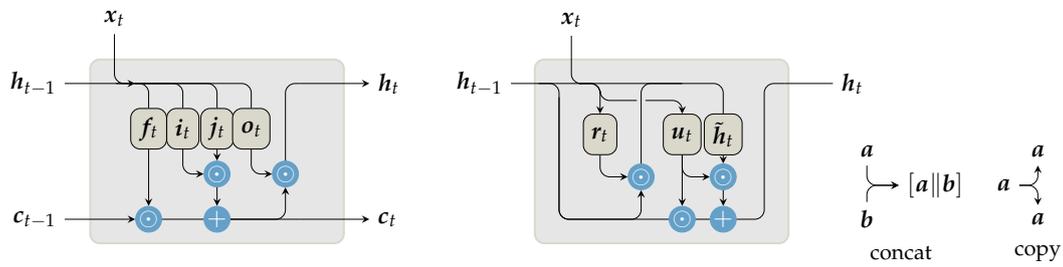

Figure 1. Schematic illustration of long short-term memory (LSTM) and gated recurrent unit (GRU) cells analogous to the cell definitions in Table 1. The cell output h_t is calculated via internal gates and, based on the current input x_t, combined with prior context information h_{t-1}, c_{t-1}. This is realized by a concatenation (concat.) of these tensors, as illustrated by merging arrows. LSTM cells are designed to separately accommodate long-term context in the internal cell state c_{t-1}, from short-term context h_{t-1}. GRU cells combine all context information in a single, but more sophisticated output h_{t-1}.

Table 1. Update formulas of the convolutional variants of standard recurrent neural networks (RNNs), long short-term memory (LSTM) cells and gated recurrent units (GRUs). A convolution between matrices a and b is denoted by $a * b$; element-wise multiplication by the Hadamard operator $a \odot b$; and concatenation on the last dimension is marked by $[a\|b]$. The activation functions sigmoid $\sigma(x)$ and tangens hyperbolicus $\tanh(x)$ are used for non-linear scaling.

Gate	Variant		
	RNN	LSTM [34]	GRU [35]
	$h_t \leftarrow x_t, h_{t-1}$	$h_t, c_t \leftarrow x_t, h_{t-1}, c_{t-1}$	$h_t \leftarrow x_t, h_{t-1}$
Forget/Reset		$f_t \leftarrow \sigma([x_t\|h_{t-1}] * W_f + 1)$	$r_t \leftarrow \sigma([x_t\|h_{t-1}] * W_r)$
Insert/Update		$i_t \leftarrow \sigma([x_t\|h_{t-1}] * W_i)$ $j_t \leftarrow \sigma([x_t\|h_{t-1}] * W_j)$	$u_t \leftarrow \sigma([x_t\|h_{t-1}] * W_u)$
Output		$o_t \leftarrow \sigma([x_t\|h_{t-1}] * W_o)$	$\tilde{h}_t \leftarrow [x_t\|r_t \odot h_{t-1}] * W$
		$c_t \leftarrow c_{t-1} \odot f_t + i_t \odot j_t$	
	$h_t \leftarrow \sigma([x_t\|h_{t-1}] * W)$	$h_t \leftarrow o_t \odot \tanh(c_t)$	$h_t \leftarrow u_t \odot h_{t-1} + (1 - u_t) \odot \tanh(\tilde{h}_t)$

To account for the more complicated design, recurrent layers are conventionally referred to as a collection of cells with a single cell representing the set of elements at one vector-index.

The common output of recurrent layers provides a many-to-many relation by generating an output vector at each observation h_t given previous context h_{t-1} and c_{t-1}, as shown in Figure 2a. However,

encoding information of the entire sequence in a many-to-one relation is favored in many applications. Following this idea, sequence-to-sequence learning, illustrated in Figure 2b, has popularized the use of the cell state vector c_T at the last-processed observation T as a representation of the entire input sequence. These encoding-decoding networks transform an input sequence of varying length to an intermediate state representation c of fixed size. Subsequently, the decoder generates a varying length output sequence from this intermediate representation. Further developments in this domain include attention schemes. These provide additional intermediate connections between encoder and decoder layers, which are beneficial for translations of longer sequences [4].

In many sequential applications, the common input form is $x_t \in \mathbb{R}^d$ with a given depth d. The output vectors $h_t \in \mathbb{R}^r$ are computed by matrix multiplication with internal weights $W \in \mathbb{R}^{(r+d)\times r}$ and r recurrent cells. However, other fields, such as image processing, commonly handle raster data $x_t \in \mathbb{R}^{h\times w\times d}$ of specific width w, height h and spectral depth d. To account for neighborhood relationships and to circumvent the increasing complexity, convolutional variants of LSTMs [36] and GRUs have been introduced. These variants convolve the input tensors with weights $W \in \mathbb{R}^{k\times k\times(r+d)\times r}$ augmented by the convolutional kernel size k, which is a hyper-parameter determining the perceptive field.
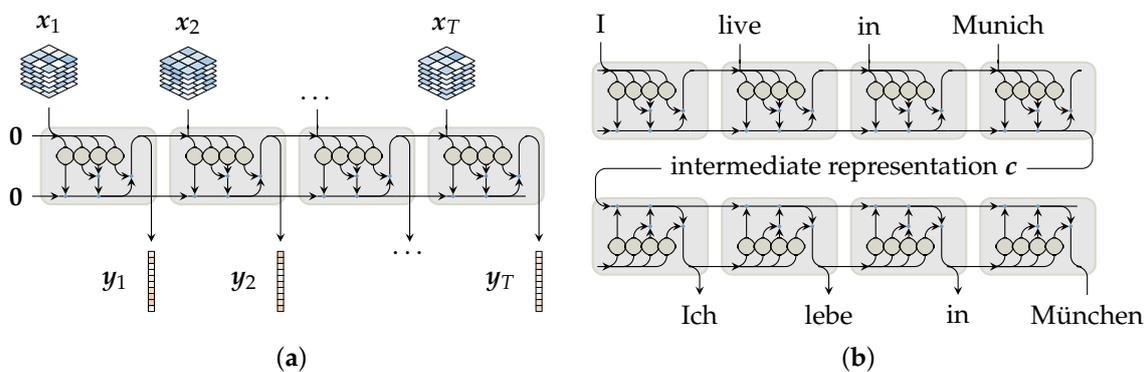

Figure 2. Illustrations of recurrent network architectures that inspired this work. The network of previous work [28] shown in (**a**) creates a prediction y_t at each observation t based on spectral input information x_t and the previous context h_{t-1}, c_{t-1}. Sequence-to-sequence networks, as shown in (**b**), aggregate sequential information to an intermediate state c_T, which is a representation of the entire series. (**a**) Network structure employed in previous work [28]; (**b**) illustration of a sequence-to-sequence network [8] as often used in neural translation tasks.

3.2. Prior Work

Given recurrent networks as popular architectures for sequential data processing, we experimented with recurrent layers for multi-temporal vegetation classification prior to this work [28]. In the conducted experiments, we used a network architecture similar to the illustration in Figure 2a. Following the input dimensions of standard recurrent layers, an input sequence $x \in \{x_0, \ldots, x_T\}$ of observations $x_t \in \mathbb{R}^d$ was introduced to the network. Based on contextual information from previous observations, a classification for each observation y_t was produced. We evaluated the effect of this information gain by comparing the recurrent network with convolutional neural networks (CNNs) and a support vector machine (SVM). Standard RNNs and LSTMs outperformed their non-sequential SVMs and CNNs counterparts. Further, we observed an increase in accuracy at sequentially later observations, which were classified with more context information available. Overall, we concluded that recurrent network architectures are well suited for the extraction of temporal features from multi-temporal EO imagery, which is consistent with other recent findings [24,26,27].

However, the experimental setup introduced some limitations regarding applicability in real-world scenarios. We followed the standard formulation of recurrent networks, which process

a *d*-dimensional input vector. This vector included the concatenated bottom-of-atmosphere (BOA) reflectances of nine pixels neighboring one point-of-interest. The point-wise classification was sufficient for quantitative accuracy evaluation, but could not produce areal classification maps. Since a class prediction was performed on every observation, we introduced additional covered classes for cloudy pixels at single images. These were derived from the scene classification of the SEN2COR atmospheric correction algorithm, which required additional preprocessing. A single representative classification for the entire time-series would have required additional post-processing to further aggregate the predicted labels for each observation. Finally, the mono-directional iterative processing introduced a bias towards last observations. With more contextual information available, later observation showed better classification accuracies compared to observations earlier in the sequence.

3.3. This Approach

To address the limitations of previous work, we redesigned and streamlined the network structure and processing pipeline. Inspired by sequence-to-sequence structures described in Section 3, the proposed network aggregates the information encoded in the cell state c_t within the recurrent cell. Since one class prediction for the entire temporal series is produced, atmospheric perturbations can be treated as temporal noise. Hence, explicitly introduced cloud-related labels are not required, which alleviates the need for prior cloud classification. Without the need for prior scene classification to obtain these classes, the performance on atmospherically uncorrected top-of-atmosphere (TOA) reflectance data can be evaluated. We further implemented convolutional recurrent cell variants, as formulated in Table 1, to process input tensors x_t of given height h, width w and depth d. Hence, the proposed network produces areal prediction maps as shown in the qualitative results Section 5.3. Finally, we introduce the input sequence in a bidirectional manner to eliminate any bias towards the later elements in the observation sequence.

Overall, we employ a bidirectional sequential encoder for the task of multi-temporal land cover classification. As Earth observation data are gathered in a periodic manner, many observations of the same area at consecutive times are available, which may contribute to the classification decision. Inspired by sequence-to-sequence models, the proposed model encodes this sequence of images into a fixed-length representation. Compared to previous work, this is an elegant way to condense the available temporal dimension without further post-processing. A classification map for each class is derived from this sequence representation. Many optical observations are covered by clouds, and prior cloud classification is often required as additional preprocessing step. As clouds do not contribute to the classification decision, these observations can be treated as temporal noise and may be potentially ignored by this encoding scheme. In Section 5.1, we investigate this by visualizing internal activation states on cloudy and non-cloudy observations.

Figure 3 presents the proposed network structure schematically. The input image sequence $x = \{x_t, \ldots, x_T\}$ of observations $x \in \mathbb{R}^{h \times w \times d}$ is passed to gated recurrent layers at each observation time t. The index T denotes the maximum length of the sequence and d the input feature depth. In practice, sequence lengths are often shorter than T, as the availability of satellite acquisitions is variable over larger scales. If less than T observations are present, sequence elements are padded with a constant value and are subsequently ignored at the iterative encoding steps. To eliminate bias towards the last observations in the sequence, the data are passed to the encoder in both sequential (seq) and reversed (rev) order. Network weights are shared between both passes. The initial cell states $c_0^{\mathrm{seq}}, c_T^{\mathrm{rev}} \in \mathbb{R}^{h \times w \times r}$ and output $h_0^{\mathrm{seq}}, h_T^{\mathrm{rev}} \in \mathbb{R}^{h \times w \times r}$ are initialized with zeros. The concatenated final states $c_T = [c_T^{\mathrm{seq}} \| c_0^{\mathrm{inv}}]$ are the representation of the entire sequence and are passed to a convolutional layer for classification. A second convolutional classification layer projects the sequence representation c_T to softmax-normalized activation maps $\hat{y}$ for n classes: $c_T \in \mathbb{R}^{h \times w \times 2r} \mapsto \hat{y} \in \mathbb{R}^{h \times w \times n}$. This layer is composed of a convolution with a kernel size of k_{class}, followed by batch normalization and a rectified

linear unit (ReLU) [37] or leaky ReLU [38] non-linear activation function. At each training step, the cross-entropy loss

$$H(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\sum_i y_i log(\hat{y}_i) \tag{1}$$

between the predicted activations $\hat{\boldsymbol{y}}$ and an one-hot representation of the ground truth labels $\boldsymbol{y}$ evaluates the prediction quality.

Tunable hyper-parameters are the number of recurrent cells r and the sizes of the convolutional kernel k_{rnn} and the classification kernel k_{class}.
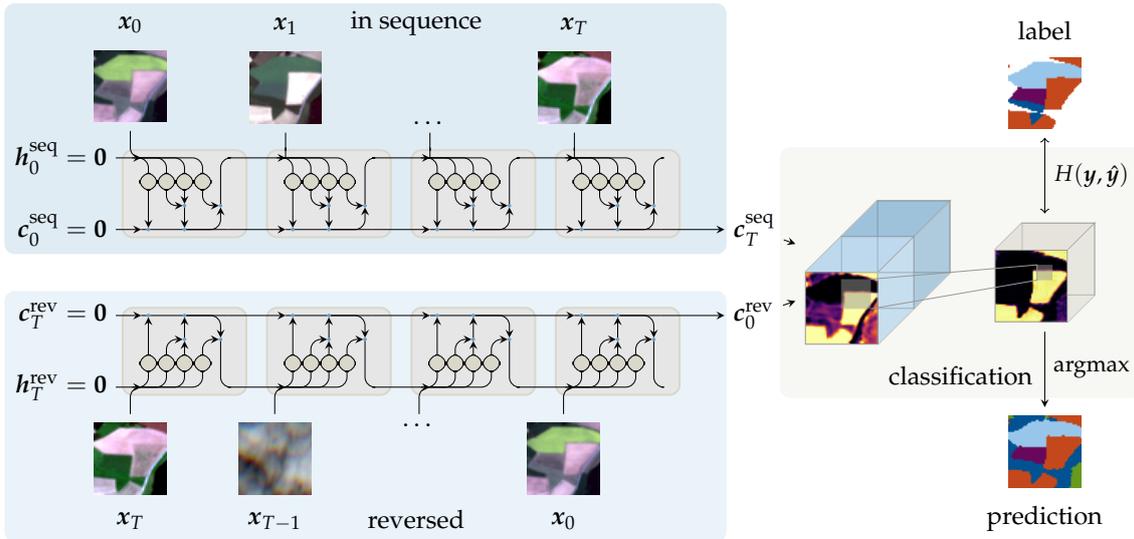

Figure 3. Schematic illustration of our proposed bidirectional sequential encoder network. The input sequence $\boldsymbol{x} \in \{\boldsymbol{x}_0, \dots, \boldsymbol{x}_T\}$ of observations $\boldsymbol{x}_t \in \mathbb{R}^{h \times w \times d}$ is encoded to a representation $\boldsymbol{c}_T = [\boldsymbol{c}_T^{\text{seq}} \| \boldsymbol{c}_0^{\text{inv}}]$. The observations are passed in sequence (seq) and reversed (rev) order to the encoder to eliminate bias towards recent observations. The concatenated representation of both passes $\boldsymbol{c}_T$ is then projected to softmax-normalized feature maps for each class using a convolutional layer.

4. Dataset

For the evaluation of our approach, we defined a large area of interest (AOI) of 102 km $\times$ 42 km north of Munich, Germany. An overview of the AOI at multiple scales is shown in Figure 4. The AOI was further subdivided into squared blocks of 3.84 km $\times$ 3.84 km (multiples of 240 m and 480 m) to ensure dataset independence while maintaining similar class distributions. These blocks were then randomly assigned to partitions for network training, hyper-parameter validation and model evaluation in a ratio of 4:1:1 similar to previous work [28]. The spatial extent of single samples $\boldsymbol{x}$ is determined by tile-grids of 240 m and 480 m. We bilinearly interpolated the 20 m and 60 m S2 bands to 10 m ground sampling distance (GSD) to harmonize the raster data dimensions. To provide additional temporal meta information, the year and day-of-year of the individual observations were added as matrices to the input tensor. Hence, the input feature depth $d = 15$ is composed of four 10 m (B4, B3, B2, B8), six 20 m (B5, B6, B7, B8A) and three 60 m (B1, B11, B12) bands combined with year and day-of-year.

With ground truth labels of two growing seasons 2016 and 2017 available, we gathered 274 (108 in 2016; 166 in 2017) Sentinel 2 products at 98 (46 in 2017; 52 in 2017) observation dates between 3 January 2016 and 15 November 2017. The obtained time series represents all available S2 products labeled with cloud coverage less than 80%. In some S2 images, we noticed a spatial offset in the scale of one pixel. However, we did not perform additional georeferencing and treated the spatial offset

as data-inherent observation noise. Overall, we relied on the geometrical and spectral reference as provided by the COPERNICUS ground segment.
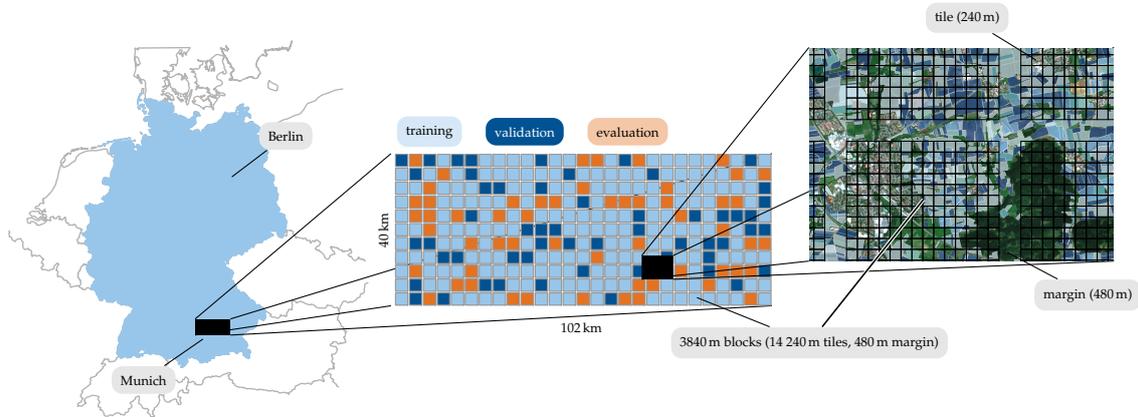

Figure 4. Area of interest (AOI) north of Munich containing 430 kha and 137 kfield parcels. The AOI is further tiled at multiple scales into datasets for training, validation and evaluation and footprints of individual samples.

Ground truth information was provided by the Bavarian Ministry of Food, Agriculture and Forestry (StMELF) in the form of geometry and semantic labels of 137 k field parcels. The crop-type is reported by farmers to the ministry as mandated by the European crop subsidy program. We selected and aggregated 17 crop-classes from approximately 200 distinct field labels, occurring at least 400 times in the AOI. With modern agriculture, centered on a few predominant crops, the distribution of classes is not uniform, as can be observed from Figure 5a. This non-uniform class distribution is generally not optimal for the classification evaluation as it skews the overall accuracy metric towards classes of high frequency. Hence, we additionally calculated kappa metrics [40] for the quantitative evaluation in Section 5.2 to compensate for unbalanced distributions.
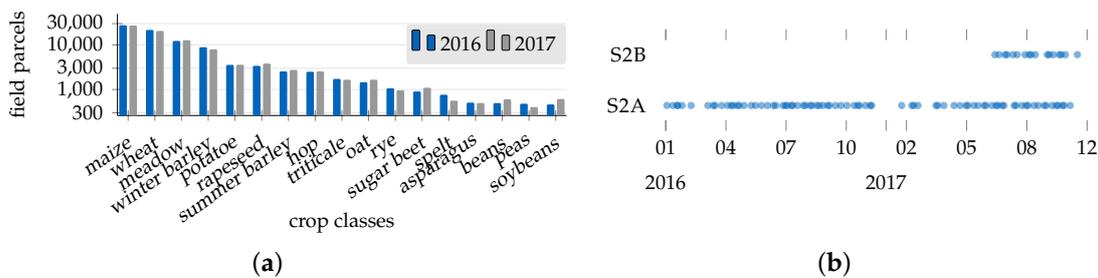

Figure 5. Information of the area of interest containing location, division schemes, class distributions and dates of acquired satellite imagery. (**a**) Non-uniform distribution of field classes in the AOI; (**b**) acquired Sentinel 2 (S2) observations of the twin satellites S2A and S2B.

5. Results

In this section, we first visualize internal state activations in Section 5.1 to gain a visual understanding of the sequential encoding process. Further findings on internal cloud masking are presented before the classification results on crop classes are quantitatively and qualitatively evaluated in Sections 5.2 and 5.3.

5.1. Internal Network Activations

In Section 3.1, we gave an overview of the functionality of recurrent layers and discussed the property of LSTM state vectors $c_t \in \mathbb{R}^{h \times w \times r}$ to encode sequential information over a series of

observations. The cell state is updated by internal gates $\boldsymbol{i}_t, \boldsymbol{j}_t, \boldsymbol{f}_t \in \mathbb{R}^{h \times w \times r}$, which in turn are calculated based on previous cell output $\boldsymbol{h}_{t-1}$ and cell state $\boldsymbol{c}_{t-1}$ (see Table 1). To assess prior assumptions regarding cloud filtering and to visually assess the encoding process, we visualized internal LSTM cell tensors for a sequence of images and show representative activations of three cells in Figure 6. The LSTM network, from which these activations are extracted, was trained on 24 px × 24 px tiles with $r = 256$ recurrent cells and $k_{\mathrm{rnn}} = k_{\mathrm{class}} = 3$ px. Additionally, we inferred the network with tiles of height h and width w of 48 px. Experiments with the input size of 24 px show similar results and are included in the Supplementary Material to this work. In the first row, a 4σ band-normalized RGB image represents the input satellite image $\boldsymbol{x}_t \in \mathbb{R}^{h=48} \times \mathbb{R}^{w=48} \times \mathbb{R}^{d=15}$ at each time frame t. The next rows show the activations of input gate $\boldsymbol{i}_t^i$, modulation gate $\boldsymbol{j}_t^i$, forget gate $\boldsymbol{f}_t^i$ and cell state $\boldsymbol{c}_t^i$ at three selected recurrent cells, which are denoted by the raised index $i \in \{3, 22, 47\}$. After iteratively processing the sequence, the final cell state $\boldsymbol{c}_{T=36}$ is used to produce activations for each class, as described in Section 3.3.

In the encoding process, the detail of structures at the cell state tensor increased gradually. This may be interpreted as additional information written to the cell state. It further appeared that the structures visible at the cell states resembled shapes, which were present in cloud-free RGB images (e.g., $\boldsymbol{c}_{t=15}^{(3)}$ or $\boldsymbol{c}_{t=28}^{(22)}$). Some cells (e.g., Cell 3 or Cell 22) changed their activations gradually over the span of multiple observations, while others (e.g., 48) changed more frequently. Forget gate $\boldsymbol{f}$ activations are element-wise multiplied with the previous cell state $\boldsymbol{c}_{t-1}$ and range between zero and one. Low values in this gate numerically reduce the cell state, which can be potentially interpreted as a change of decision. The input $\boldsymbol{i}$ and modulation gate $\boldsymbol{j}$ control the degree of new information written to the cell state. While the input gate is scaled between zero and one, the modulation gate $\boldsymbol{j} \in [-1, 1]$ determines the sign of change. In general, we found the activity of a majority of cells (e.g., Cell 3 or Cell 22) difficult to associate with distinct events in the current input. However, we assumed that classification-relevant features were expressed as a combination of cell activations similar to other neural network approaches. Nevertheless, we could identify a proportionally small number of cells, in which the shape of clouds visible in the image was projected on the internal state activations. One of these was cell $i = 47$. For cloudy observations, the input gate approached zero either over the entire tile (e.g., $t = \{10, 18, 19, 36\}$) or over patches of cloudy pixels (e.g., $t = \{11, 13, 31, 33\}$). At some observation times (e.g., $t = \{13, 31, 32\}$), the modulation gate $\boldsymbol{j}_t^{(47)}$ additionally changed the sign.

In a similar fashion, Karpathy [41] evaluated cell activations for the task of text processing. He could associate a small number of cells with a set of distinct tasks, such as monitoring the lengths of a sentence or maintaining a state-flag for text inside and outside of brackets.

Summarizing this experiment, the majority of cells showed increasingly detailed structures when new information was provided in the input sequence. It is likely that the grammar of crop-characteristic phenological changes was encoded in the network weights, and we suspect that a certain amount of these cells was sensitive to distinct events relevant for crop identification. However, these events may be encoded in multiple cells and were difficult to visually interpret. A small set of cells could be visually associated with individual cloud covers and may be used for internal cloud masking. Based on these findings, we are confident that our network has learned to internally filter clouds without explicitly introducing cloud-related labels.
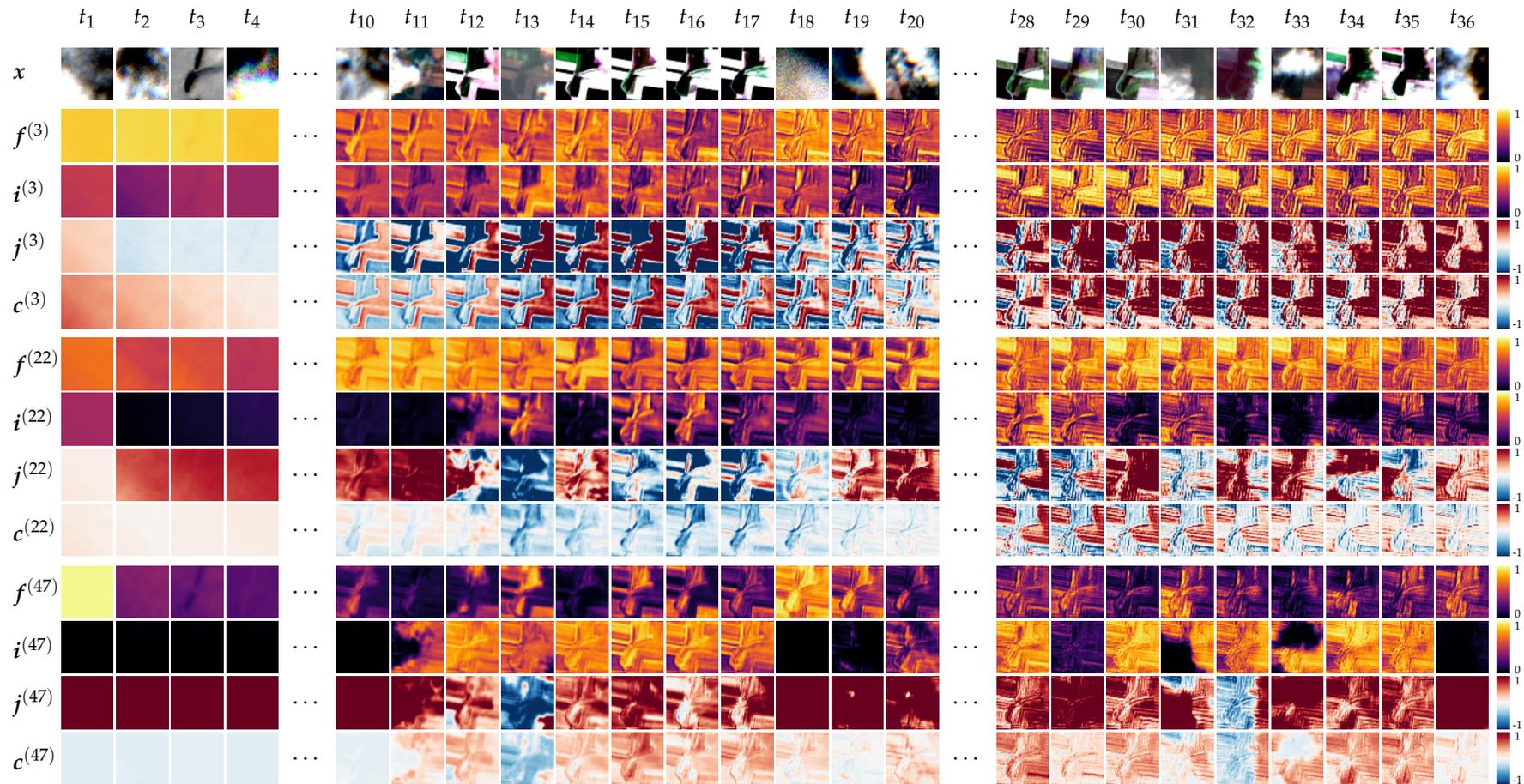
Figure 6. Internal LSTM cell activations of input gate $i^{(i)}$, forget gate $f^{(i)}$, modulation gate $j^{(i)}$ and cell state $c^{(i)}$ at three (of $r = 256$) selected cells $i \in \{3, 22, 47\}$ given the current input x_t over the sequence of observations $t = \{1, \ldots, 36\}$. The detail of features at the cell states increased gradually, which indicated the aggregation of information over the sequence. While most cells likely contribute to the classification decision, only some cells are visually interpretable with regard to the current input x_t. One visually-interpretable cell $i = 47$ has learned to identify cloud, as input and modulation gates show different activation patterns on cloudy and non-cloudy observations.

5.2. Quantitative Classification Evaluation

For the quantitative evaluation of our approach, we trained networks with bidirectional convolutional LSTM and GRU cells with $r \in \{128, 256\}$ recurrent cells. Kernel sizes of $k_{\text{rnn}} = k_{\text{class}} = 3$ were used for the evaluation since previous tests with larger kernel sizes showed similar accuracies. For these initial experiments, we predominantly tested network variants with $r = 128$ recurrent cells, as these networks could be trained within a reasonable time frame. We decided to use networks with $r = 256$ recurrent cells for the final accuracy evaluation, as we found that these variants achieved slightly better results in prior tests. The convolutional GRU and LSTM networks were trained on a P100 GPU for 60 epochs (3.51 Mio 24 px × 24 px tiles seen) and took 58 h and 51 h, respectively. However, reasonable accuracies were achieved within the first twelve hours, and further training increased the accuracies on validation data only marginally. At each training step, a subset of 30 observations was randomly sampled from all available 46 (2016) and 52 (2017) observations to randomize the sequence while the sequential order was maintained. For all our tests, the performance of LSTM and GRU networks was similar. The fewer weights of GRU cells, however, allowed using a slightly larger batch size of 32 samples compared to 28 samples of the LSTM variant. This led to a seven-hour faster training compared to the LSTM variant.

For these reasons, we decided to report evaluation results of the GRU network in Table 2. Precision and recall are common accuracy measures that normalize the sum of correctly-predicted samples with the total number of predicted and reference samples of a given class, respectively. These measures are equivalent to user's and producer's accuracies and inverse to errors of commission and omission, which are popular metrics in the remote sensing community. We further calculated the f-measure as the harmonic average of precision and recall and the overall accuracy as the sum of correctly-classified samples normalized by the total number of samples. These metrics weight each sample equally. This introduces a bias towards frequent classes in the dataset, such as maize or wheat. To compensate for the non-uniform class distribution, we additionally report the conditional [42] and overall kappa [40] coefficients, which are normalized by the probability of a hypothetical correct classification by chance. The kappa coefficient κ is a measure of agreement and typically ranges between $\kappa = 0$ for no and $\kappa = 1$ for complete agreement. McHugh [43] provides an interpretative table in which values $0.4 \leq \kappa < 0.6$ are considered 'weak', values $0.6 \leq \kappa < 0.8$ 'moderate', $0.8 \leq \kappa \leq 0.9$ considered 'strong' and values beyond 0.9 'almost perfect'.

The provided table of accuracies shows precision, recall, f-measure and the conditional kappa coefficient for each class over the two evaluated seasons. Furthermore, overall accuracy and overall kappa coefficients indicate the quality for the classification and report good accuracies. The pixel-averaged achieved precision, recall and f-score accuracies were consistent and ranged between 89.3% and 89.9%. The kappa coefficients of 0.870 and overall accuracies of 89.7% and 89.5% show similar consistency. While these classification measures reported good performances, the class-wise accuracies varied largely between 41.5% (peas) and 96.8% (maize). For better visibility, we emphasized the best and worst metrics by boldface. The conditional kappa scores are similarly variable and range between 0.414 (peas) and 0.957 (rapeseed).

Frequent classes (e.g., maize, meadow) have been in general more confidently classified than less frequent classes (e.g., peas, summer oat, winter spelt, winter triticale). Nonetheless this relation has exceptions. The least frequent class, peas, performed relatively well on data of 2016, and other less frequent classes, such as asparagus or hop, showed good performances despite their underrepresentation in the dataset.

To investigate the causes of the varying accuracies, we calculated confusion matrices for both seasons as shown in Figure 7. These error matrices are two-dimensional histograms of classification samples aggregated by the class prediction and ground truth reference. To account for the non-uniform class distribution, the absolute number of samples for each row-column pair is normalized. We decided to normalize the confusion matrices by row to obtain recall (producer's) accuracies, due to their direct relation to available ground truth labels. The diagonal elements of the matrices represent

correctly-classified samples with values equivalent to Table 2. Structures outside the diagonal indicate systematic confusions between classes and may give insight into the reasoning behind varying classification accuracies.
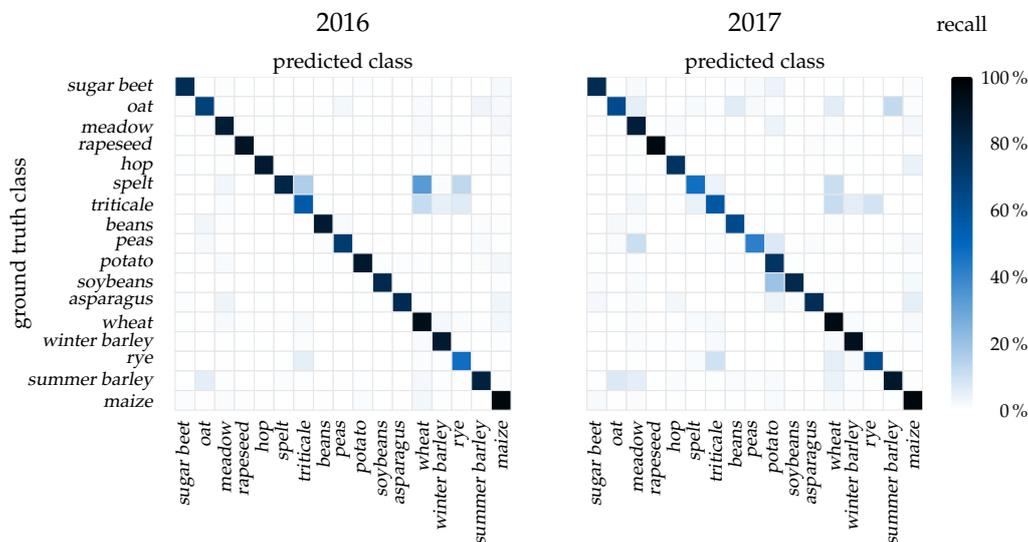

Figure 7. Confusion matrix of the trained convolutional GRU network on data of the seasons 2016 and 2017. While the confusion of some classes was consistent over both seasons (e.g., winter triticale to winter wheat), other classes are classified at different accuracies for consecutive years (e.g., winter barley to winter spelt).

Table 2. Pixel-wise accuracies of the trained convolutional GRU sequential encoder network after training over 60 epochs on data of both growth seasons. The conditional kappa metrics [42] for each class and the overall kappa [40] measure are given for both growth seasons. The best and worst metrics are emphasized by boldface.

Class	Year									
	2016					2017				
	Precision (User's Acc.)	Recall (Prod.Acc.)	*f*-Meas.	Kappa	# of Pixels	Precision (User's Acc.)	Recall (Prod. Acc.)	*f*-Meas.	Kappa	# of Pixels
sugar beet	94.6	77.6	85.3	0.772	59 k	89.2	78.5	83.5	0.779	94 k
oat	86.1	67.8	75.8	0.675	36 k	63.8	62.8	63.3	0.623	38 k
meadow	90.8	85.7	88.2	0.845	233 k	88.1	85.0	86.5	0.837	242 k
rapeseed	95.4	90.0	92.6	0.896	125 k	**96.2**	95.9	96.1	**0.957**	114k
hop	**96.4**	87.5	91.7	0.873	51 k	92.5	74.7	82.7	0.743	53 k
spelt	**55.1**	81.1	65.6	0.807	38 k	75.3	46.7	57.6	0.463	31 k
triticale	69.4	55.7	61.8	0.549	65 k	62.4	57.2	59.7	0.563	64 k
beans	92.4	87.1	89.6	0.869	27 k	92.8	63.2	75.2	0.630	28 k
peas	93.2	70.7	80.4	0.706	9 k	**60.9**	**41.5**	49.3	**0.414**	6 k
potato	90.9	88.2	89.5	0.876	126 k	95.2	73.8	83.1	0.728	140 k
soybeans	97.7	79.6	87.7	0.795	21 k	75.9	79.9	77.8	0.798	26 k
asparagus	89.2	78.8	83.7	0.787	20 k	81.6	77.5	79.5	0.773	19 k
wheat	87.7	93.1	90.3	0.902	806 k	90.1	95.0	92.5	0.930	783 k
winter barley	95.2	87.3	91.0	0.861	258 k	92.5	92.2	92.4	0.915	255 k
rye	85.6	**47.0**	**60.7**	**0.466**	43 k	76.7	61.9	68.5	0.616	30 k
summer barley	87.5	83.4	85.4	0.830	73 k	77.9	88.5	82.9	0.880	91 k
maize	91.6	**96.3**	**93.9**	**0.944**	919 k	92.3	**96.8**	94.5	0.953	876 k
weight.avg	**89.9**	**89.7**	**89.5**			**89.5**	**89.5**	**89.3**		
	Overall Accuracy		Overall Kappa			Overall Accuracy		Overall Kappa		
	89.7		**0.870**			**89.5**		**0.870**		

Some crops likely share common spectral or phenological characteristics. Hence, we expected some symmetric confusion between classes, which would be expressed as diagonal symmetric confusions consistent in both years. Examples of this were triticale and rye or oat and summer

barley. However, these relations were not frequent in the dataset, which indicates that the network had sufficient capacity to separate the classes by provided features. In some cases, one class may share characteristics with another class. This class may be further distinguished by additional unique features, which would be expressed by asymmetric confusions between these two classes in both seasons. Relations of this type were more dominantly visible in the matrices and included confusions between barley and triticale, triticale and spelt or wheat confused with triticale and spelt. These types of confusion were consistent over both seasons and may be explained by a spectral or phenological similarity between individual crop-types.

More dominantly, many confusions were not consistent over the two growing seasons. For instance, confusions occurring only in the 2017 season were soybeans with potato or peas with meadow and potato. Since the cultivated crops are identical in these years and the class distributions were consistent, seasonally-variable factors were likely responsible for these relations. As reported in Table 2, peas have been classified well in 2016, but poorly in 2017, due to the aforementioned confusions with meadow and potato. These results indicate that external and not crop-type-related factors had a negative influence on classification accuracies, which appeared unique to one season. One of these might be the variable onset of phenological events, which are indirectly observed by the change of reflectances by the sensors. These events are influenced by local weather and sun exposure, which may vary over large regional scales or multiple years.

5.3. Qualitative Classification Evaluation

For the qualitative evaluation, we used the same network structure as in the previous section. We inferred the network with 48 px tiles from the evaluation dataset of 2017 for better visibility. In Figure 8, a series of good (A–D) and bad (E,F) classification examples are shown. The first column represents the input sequence x as band-normalized RGB images from one selected cloud-free observation $x_{RGB,t}$. Further columns show the available ground truth labels y, predictions $\hat{y}$ and the cross-entropy loss $H(y, \hat{y})$. Additionally, four selected softmax-normalized class activations are displayed in the last columns. These activations can be interpreted as classification confidences for each class. The prediction map contains the index of the most activated class at each pixel, which may be interpreted as the class of highest confidence. The cross-entropy loss is the measure the agreement between the one-hot representation of the ground truth labels and the activations per class. It is used as the objective function, as network training indicates disagreement between ground truth and prediction even if the final class prediction is correct. This relation can be observed in fields of several examples, such as peas in Example A, spelt in Example B and oat in Example C. However, most classifications for these examples were accurate, which is expressed by well-defined activation maps.

Often, classifiers use low-pass filters in the spatial dimensions to compensate for high-frequent noise. These filters typically limit the ability to classify small objects. To evaluate to what degree the network has learned to apply low-pass filtering, we show a tile with a series of narrow fields in Example D. Two thin wheat and maize fields have been classified correctly. However, some errors occurred on the southern end of an adjacent potato field, as indicated by the loss map. It appears that the network was able to resolve high-frequency spatial changes and did not apply smoothing of the class activations, as in Example F.

Two misclassified fields are shown in Example E. The upper wheat field has been confidently misclassified to summer barley. Underneath, the classification of a second rye field was uncertain between rye, wheat and triticale. While triticale, as the least activated class, was not present in the prediction map, the mixture of rye and wheat is visible in the class predictions.

Example F shows a mostly misclassified tile. Only a few patches of meadow and winter barley were correctly predicted. The activations of these classes were, compared to previous examples, generally more blurred and of lower amplitude. Similar to Example D, the most activated classes are also the most frequent in the dataset. In fact, the entire region around the displayed tile seemed to be classified poorly. This region was located on the northwest border of the AOI. Further examination showed that

for this region, fewer satellite images were available. The lack of temporal information likely explains the poor classification accuracies. However, this example illustrates that the class activations give an indication of the classification confidence independent of the ground truth information.
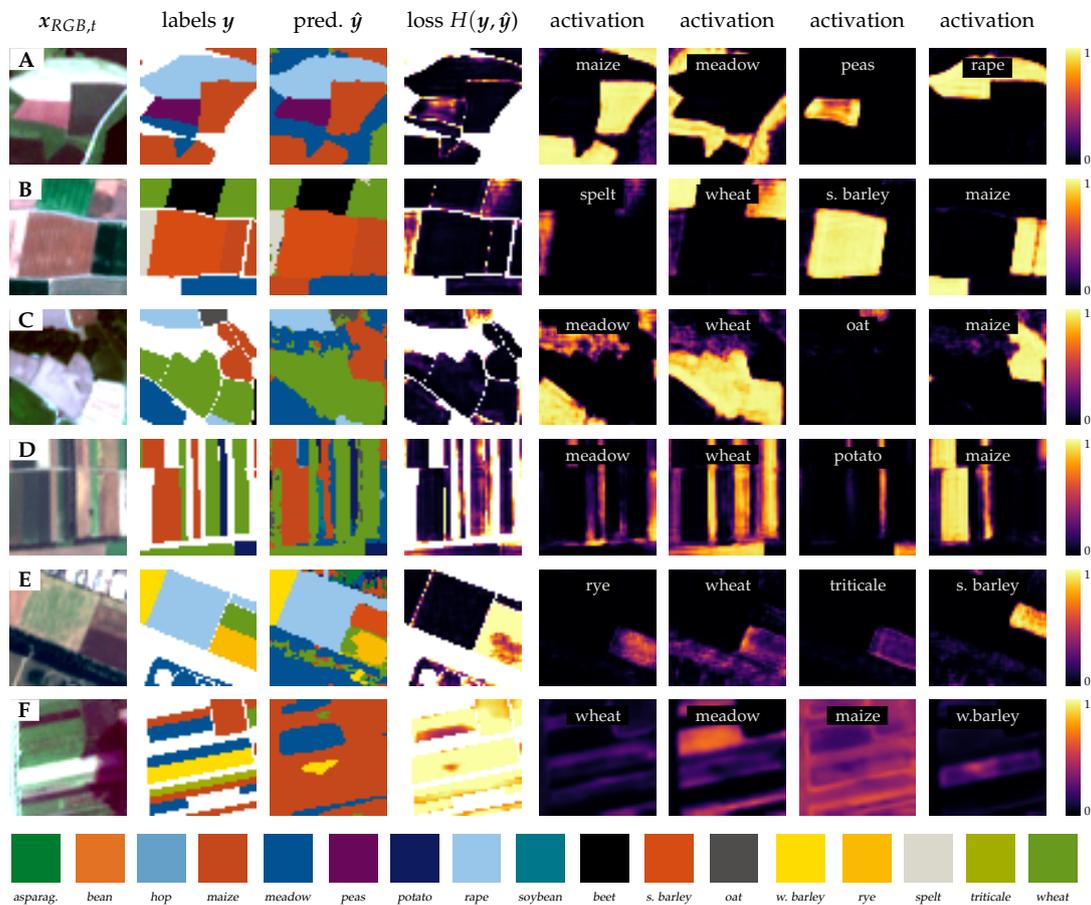

Figure 8. Qualitative results of the convolutional GRU sequential encoder. Examples (**A**–**D**) show good classification results. For Example (**E**) the network misclassified one maize parcel with high confidence, which is indicated by incorrect, but well-defined activations. In a second field, the class activations reveal a confusion between wheat, meadow and maize. For Example (**F**), most pixels are misclassified. However, the class activations show uncertainty in the classification decision.

6. Discussion

In this section, we compare our approach with other multi-temporal classifications. Unfortunately, to the best of our knowledge, no multi-temporal benchmark dataset is available to compare remote sensing approaches on equal footing. Nevertheless, we provide some perspective of the study domain by gathering multi-temporal crop classification approaches in Table 3 and categorizing these by their applied methodology and achieved overall accuracy. However, the heterogeneity of data sources, the varying extents of their evaluated areas and the number of classes used in these studies impedes a numerical comparison of the achieved accuracies. Despite this, we hope that this table will provide an overview of the state-of-the-art in multi-temporal crop identification.

Earth observation (EO) data are acquired in periodic intervals at high spatial resolutions. From an information theoretical perspective, utilizing additional data should lead to better classification performance. However, the large quantity of data requires methods that are able to process this information and are robust with regard to observation noise. Optimally, these approaches are scalable with minimal supervision so that data of multiple years can be included over large regions. Existing approaches in multi-temporal EO tasks often use multiple separate processing steps, such as preprocessing, feature

extraction and classification, as summarized by Ünsalan and Boyer [44]. Generally, these steps require manual supervision or the selection of additional parameters based on region-specific expert knowledge, a process that impedes applicability at large scales. The cost of data acquisition is an additional barrier, as multiple and potentially expensive satellite images are required. Commercial satellites, such as RapidEye (RE), Satellite Pour l'Observation de la Terre (SPOT) or QuickBird (QB), provide images at excellent spatial resolution. However, predominantly inexpensive sensors, such as Landsat (LS), Sentinel 2 (S2), Moderate-resolution Imaging Spectroradiometer (MODIS) or Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), can be applied at large scales, since the decreasing information gain of additional observations must justify image acquisition costs. Many approaches use spectral indices, such as normalized difference vegetation index (NDVI), normalized difference water index (NDWI) or enhanced vegetation index (EVI), to extract statistical features from vegetation-related signals and are invariant to atmospheric perturbations. Commonly, decision trees (DTs) or random forests (RFs) are used for classification. The exclusive use of spectral indices simplifies the task of feature extraction. However, these indices utilize only a small number of available spectral bands (predominantly blue, red and near-infrared). Thus, methods that utilize all reflectance measurements, either at top-of-atmosphere (TOA), or atmospherically-corrected to bottom-of-atmosphere (BOA), are favorable, since all potential spectral information can be extracted.

Table 3. Overview of recent approaches for crop classification.

Approach	Details					
	Sensor	Preprocessing	Features	Classifier	Accuracy	# of Classes
this work	S2	none	TOA reflect.	ConvRNN	90	17
Rußwurm and Körner [28], 2017	S2	atm. cor.(SEN2COR)	BOA reflect.	RNN	74	18
Siachalou et al. [15], 2015	LS, RE	geometric correction, image registration	TOA reflect.	HMM	90	6
Hao et al. [13], 2015	MODIS	image reprojection, atm. cor. [45]	statistical phen.features	RF	89	6
Conrad et al. [12], 2014	SPOT, RE, QB	segmentation, atm. cor. [45]	vegetation indices	OBIA + RF	86	9
Foerster et al. [10], 2012	LS	phen. normalization, atm. cor. [45]	NDVI statistics	DT	73	11
Peña-Barragán et al. [14], 2011	ASTER	segmentation, atm. cor. [46]	vegetation indices	OBIA+ DT	79	13
Conrad et al. [11], 2010	SPOT ASTER	segmentation, atm. cor. [45]	vegetation indices	OBIA + DT	80	6

In general, a direct numerical comparison of classification accuracies is difficult, since these are dependent on the number of evaluated samples, the extent of evaluated area and the number of classified categories. Nonetheless, we compare our method with the approaches of Siachalou et al. [15] and Hao et al. [13] in detail since their achieved classification accuracies are on a similar level as ours. Hao et al. [13] used an RF classifier on phenological features, which were extracted from NDVI and NDWI time series of MODIS data. Their results demonstrate that good classification accuracies with hand-crafted feature extraction and classification methods can be achieved if data of sufficient temporal resolution are available. However, the large spatial resolution (500 m) of the MODIS sensor limits the applicability of this approach to areas of large homogeneous regions. On a smaller scale, Siachalou et al. [15] report good levels of accuracy on small fields. For this, they used a hidden Markov models (HMMs) with a temporal series of four LS images combined with one single RapidEye

(RE) image for field border delineation. Methodologically, HMMs and conditional random fields (CRFs) [16] are closer to our approach since the phenological model is approximated with an internal chain of hidden states. However, these methods might not be applicable for long temporal series, since Markov-based approaches assume that only one previous state contains classification-relevant information.

Overall, this comparison shows that our proposed network can achieve state-of-the-art classification accuracy with a comparatively large number of classes. Furthermore, the S2 data of non-atmospherically-corrected TOA values can be acquired easily and does not require further preprocessing. Compared to previous work, we were able to process larger tiles by using convolutional recurrent cells with only a single recurrent encoding layer. Moreover, we neither required atmospheric correction, nor additional cloud classes, since one classification decision is derived from the entire sequence of observations.

7. Conclusions

In this work, we proposed an automated end-to-end approach for multi-temporal classification, which achieved state-of-the-art accuracies in crop classification tasks with a large number of crop classes. Furthermore, the reported accuracies were achieved without radiometric and geometric preprocessing. The trained and inferred data were atmospherically uncorrected and contained clouds. In traditional approaches, multi-temporal cloud detection algorithms utilize the sudden positive change in reflectivity of cloudy pixels and achieve better results than other traditional mono-temporal remote sensing classifiers [47]. Results of this work indicate that cloud masking can be learned jointly together with classification. By visualizing internal gate activations in our network in Section 5.1, we found evidence that some recurrent cells were sensitive to cloud coverage. These cells may be used by the network to internally mask cloudy pixels similar to an external cloud filtering algorithm.

In Sections 5.2 and 5.3, we further evaluated the classification results quantitatively and qualitatively. Based on several findings, we derived that the network has approximated a discriminative crop-specific phenological model based on a raw series of TOA S2 observations. Further inspection revealed that some crops were inconsistently classified in both growing seasons. This may be caused by seasonally-variable environmental conditions, which may have been implicitly integrated into the encoded phenological model. We employed our network for the task crop classification since vegetative classes are well characterized by their inherently temporal phenology. However, the network architecture is methodologically not limited to vegetation modeling and may be employed for further tasks, which may benefit from the extraction of temporal features. We hope that our results encourage the research community to utilize the temporal domain for their applications. In this regard, we publish the TENSORFLOWsource code of our network along with the evaluations and experiments from this work.

References

1. Zhang, L.; Zhang, Q.; Du, B.; Huang, X.; Tang, Y.Y.; Tao, D. Simultaneous Spectral-Spatial Feature Selection and Extraction for Hyperspectral Images. *IEEE Trans. Cybern.* **2018**, *48*, 16–28.

2. Odenweller, J.B.; Johnson, K.I. Crop identification using Landsat temporal-spectral profiles. *Remote Sens. Environ.* **1984**, *14*, 39–54.

3. Reed, B.C.; Brown, J.F.; VanderZee, D.; Loveland, T.R.; Merchant, J.W.; Ohlen, D.O. Measuring Phenological Variability from Satellite Imagery. *J. Veg. Sci.* **1994**, *5*, 703–714.

4. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473v7.

5. Rush, A.; Chopra, S.; Weston, J. A Neural Attention Model for Sentence Summarization. *arXiv* **2017**, arXiv:1509.00685v2

6. Shen, S.; Liu, Z.; Sun, M. Neural Headline Generation with Minimum Risk Training. *arXiv* **2016**, arXiv:1604.01904v1.

7. Nallapati, R.; Zhou, B.; dos Santos, C.N.; Gulcehre, C.; Xiang, B. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. *arXiv* **2016**, arXiv:1602.06023v5.

8. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215v3.

9. Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. *Adv. Neural Inf. Process. Syst.* **2015**, *1*, 557–585.

10. Foerster, S.; Kaden, K.; Foerster, M.; Itzerott, S. Crop type mapping using spectral-temporal profiles and phenological information. *Comput. Electron. Agric.* **2012**, *89*, 30–40.

11. Conrad, C.; Fritsch, S.; Zeidler, J.; Rücker, G.; Dech, S. Per-Field Irrigated Crop Classification in Arid Central Asia Using SPOT and ASTER Data. *Remote Sens.* **2010**, *2*, 1035–1056.

12. Conrad, C.; Dech, S.; Dubovyk, O.; Fritsch, S.; Klein, D.; Löw, F.; Schorcht, G.; Zeidler, J. Derivation of temporal windows for accurate crop discrimination in heterogeneous croplands of Uzbekistan using multitemporal RapidEye images. *Comput. Electron. Agric.* **2014**, *103*, 63–74.

13. Hao, P.; Zhan, Y.; Wang, L.; Niu, Z.; Shakir, M. Feature Selection of Time Series MODIS Data for Early Crop Classification Using Random Forest: A Case Study in Kansas, USA. *Remote Sens.* **2015**, *7*, 5347–5369.

14. Peña-Barragán, J.M.; Ngugi, M.K.; Plant, R.E.; Six, J. Object-based crop identification using multiple vegetation indices, textural features and crop phenology. *Remote Sens. Environ.* **2011**, *115*, 1301–1316.

15. Siachalou, S.; Mallinis, G.; Tsakiri-Strati, M. A hidden markov models approach for crop classification: Linking crop phenology to time series of multi-sensor remote sensing data. *Remote Sens.* **2015**, *7*, 3633–3650.

16. Hoberg, T.; Rottensteiner, F.; Feitosa, R.Q.; Heipke, C. Conditional random fields for multitemporal and multiscale classification of optical satellite imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 659–673.

17. Zhang, L.; Zhang, L.; Du, B. Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40.

18. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36.

19. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sens.* **2015**, *7*, 14680–14707.

20. Scott, G.J.; England, M.R.; Starms, W.A.; Marcum, R.A.; Davis, C.H. Training Deep Convolutional Neural Networks for Land-Cover Classification of High-Resolution Imagery. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 549–553.

21. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep Supervised Learning for Hyperspectral Data Classification through Convolutional Neural Networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (GARSS), Milan, Italy, 26–31 July 2015; pp. 4959–4962.

22. Castelluccio, M.; Poggi, G.; Sansone, C.; Verdoliva, L. Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. *arXiv* **2015**, arXiv:1508.00092.

23. Lyu, H.; Lu, H.; Mou, L. Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection. *Remote Sens.* **2016**, *8*, 506.

24. Jia, X.; Khandelwal, A.; Nayak, G.; Gerber, J.; Carlson, K.; West, P.; Kumar, V. Incremental Dual-memory LSTM in Land Cover Prediction. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 867–876.

25. Mou, L.; Bruzzone, L.; Zhu, X.X. Learning Spectral-Spatial-Temporal Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery. *arXiv* **2018**, arXiv:1803.02642v1.

26. Braakmann-Folgmann, A.; Roscher, R.; Wenzel, S.; Uebbing, B.; Kusche, J. Sea Level Anomaly Prediction using Recurrent Neural Networks. *arXiv* **2017**, arXiv:1710.07099v1.

27. Sharma, A.; Liu, X.; Yang, X. Land Cover Classification from Multi-temporal, Multi-spectral Remotely Sensed Imagery using Patch-Based Recurrent Neural Networks. *arXiv* **2017**, arXiv:1708.00813v1.

28. Rußwurm, M.; Körner, M. Temporal Vegetation Modelling using Long Short-Term Memory Networks for Crop Identification from Medium-Resolution Multi-Spectral Satellite Images. In Proceedings of the IEEE/ISPRS Workshop on Large Scale Computer Vision for Remote Sensing Imagery (EarthVision), Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA , 21–26 July 2017.

29. Graves, A.; Wayne, G.; Danihelka, I. Neural Turing Machines. *arXiv* **2014**, arXiv:1410.5401v2.

30. Siegelmann, H.; Sontag, E. On the Computational Power of Neural Nets. *J. Comput. Syst. Sci.* **1995**, *50*, 132–150.

31. Rafal, J.; Wojciech, Z.; Ilya, S. An Empirical Exploration of Recurrent Network Architectures. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 7, pp. 2342–2350.

32. Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J. Gradient flow in recurrent nets: The difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Networks*; IEEE Press: New York, NY, USA, 2001; pp. 237–243.

33. Yoshua, B.; Patrice, S.; Paolo, F. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166.

34. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780.

35. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078v3.

36. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, *1*, 802–810.

37. Hahnloser, R.; Sarpeshkar, R.; Mahowald, M.A.; Douglas, R.J.; Seung, H.S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* **2000**, *405*, 947–951.

38. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Proc. Int. Conf. Mach. Learn.* **2013**, *28*, 6.

39. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980v9.

40. Cohen, J. A coefficient of agreeement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46.

41. Karpathy, A.; Johnson, J.; Fei-Fei, L. Visualizing and Understanding Recurrent Networks. *arXiv* **2015**, arXiv:1506.02078.

42. Fung, T.; Ledrew, E. The Determination of Optimal Threshold Levels for Change Detection Using Various Accuracy Indices. *Photogramm. Eng. Remote Sens.* **1988**, *54*, 1449–1454.

43. McHugh, M.L. Interrater reliability: the kappa statistic. *Biochem. Med.* **2012**, *22*, 276–282.

44. Ünsalan, C.; Boyer, K.L. Review on Land Use Classification. In *Multispectral Satellite Image Understanding: From Land Classification to Building and Road Detection*; Springer: London, UK, 2011; pp. 49–64.

45. Richter, R. A spatially adaptive fast atmospheric correction algorithm. *Int. J. Remote Sens.* **1996**, *17*, 1201–1214.

46. Matthew, M.W.; Adler-Golden, S.M.; Berk, A.; Richtsmeier, S.C.; Levine, R.Y.; Bernstein, L.S.; Acharya, P.K.; Anderson, G.P.; Felde, G.W.; Hoke, M.P. Status of Atmospheric Correction using a MODTRAN4-Based Algorithm. In Proceedings of the SPIE Algorithms for Multispectral, Hyperspectral, and Ultra-Spectral Imagery VI, Orlando, FL, USA, 16–20 April 2000; pp. 199–207.

47. Hagolle, O.; Huc, M.; Villa Pascual, D.; Dedieu, G. A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENuS, LANDSAT and SENTINEL-2 images. *Remote Sens. Environ.* **2010**, *114*, 1747–1755.