UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Kyrylo Medianovskyi

Exploring DeepSense Neural Network Architecture for Farming Events Detection

Master's Thesis (30 ECTS)

Supervisor: Amnir Hadachi, PhD.

Tartu 2020

Exploring DeepSense Neural Network Architecture for Farming Events Detection

Abstract:

Nowadays satellite imagery became widely available and found to be applicable in a range of different areas. Agriculture is one of those domains. With the help of imagery data there is a set of processes that can be automatized. Thousands of people across the European Union are involved in field inspection. They are checking the crop types and take a record of mowing events that happen on the parcels. Estonia has a relatively high level of cloud coverage and rains during a vegetation season. That leads to interruptions and noises in satellite imagery data. A noise tolerating automated mowing event detection system is required.

For this thesis Sentinel-1 coherence for VV and VH polarisation together with Sentinel-2 normalized difference vegetation index were chosen as the main features to build a mowing event recognition system. The architecture DeepSense is implemented and evaluated as a mowing event detection mechanism. The system was trained on Estonia 2018 labeled data containing information about over 1700 fields. An optimal configuration of hyper-parameters was obtained based on experiments with the architecture.

Proposed modification of the DeepSense framework allowed to reach 94% event accuracy and 93% end of season event accuracy obtained from 5-fold cross-validation. The DeepSense implementation allowed to outperform a purely convolutional model based on the end of season accuracy metric (93% against 90%). The proposed architecture can be adopted for the mowing event detection tasks.

Keywords:

Sentinel-1, Sentinel-2, DeepSense, farming events detection, Convolutional neural networks, Recursive neural networks.

CERCS: P170. Computer science, numerical analysis, systems, control.

DeepSense närvivõrgu arhitektuuri uurimine põllumajandusürituste tuvastamiseks

Lühikokkuvõte:

Tänapäeval sai satelliidi pilte laialt saadaval ja see leiab rakendust erinevates valdkondades. Põllumajandus on üks neist valdkondadest. Visuaalsete andmete abil on võimalik automatiseerida terve hulk erinevaid protsesse. Tuhandeid inimesi üle Euroopa Liidu on seotud põldtunnustamisega. Nad kontrollivad põllukultuuride tüüpe ja arvestavad niitmiseid, mis juhtub kruntidel. Eestis on vegetatsiooni perioodil esineb suhteliselt pilvine ja vihmane ilm. See põhjustab katkestusi ja müra satelliitpiltide andmetes. Sellega on vaja arendada müra taluvat automatiseeritut niitmise tuvastamise süsteemi.

Selle teosega niitmise tuvastamise süsteemi ehitamiseks said valitud põhitunnused. Nende hulka kuulub VV ja VH polarisatsiooni sidusus Sentinel-1 koos Sentinel-2 normaliseeritud erinevuse taimestiku indeksiga. DeepSense arhitektuuri on rakendatud ja hinnatud niitmise sündmuse avastamise mehhanismina. Süsteem oli koolitatud Eestis 2018 aastal märgistatud andmetega, mis sisaldavad infot üle 1700 põllu kohta. Optimaalse konfiguratsiooni hüperparameetrid on saadud katsetamise ajal rakendades vastava arhitektuuri.

Kavandatud muudatused DeepSense raamistikus võimaldasid saavutada 94% sündmuste täpsust hooaja alguses ja 93% hooaja lõpus, sündmuste täpsust saadud 5-kordse ristvalidatsiooniga. DeepSense rakendamine võimaldas edestada puhta konvolutsionaalse mudeli, mis põhineb hooaja lõpu täpsusmõõdiku peal (93% vs 90%). Väljapakutud arhitektuur saab kasutusele võtta niitmise sündmuste avastamiseks.

Võtmesõnad:

Sentinel-1, Sentinel-2, DeepSense, Põllumajanduslike sündmuste avastamine, Konvolutionaalsed neuraalvõrgustikud, Rekursiivsed neuraalvõrgustikud.

CERCS: P170. Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria).

Acknowledgement

I would like to express my gratitude to Dr. Amnir Hadachi who was supervising me during my Master's studies. I wish to remark his strong support, wonderful time management, and high level of responsiveness during my work. He gave me advice at the moments of uncertainty and, at the same time, he let me continue work making my own decisions independently. It was extremely valuable to get the assessment of my results during intermediate stages.

Also I would like to thank Dr. Kaupo Voormansik and Kappazeta team for guidance and hints during the development stage of this thesis. I appreciate the opportunities Kappazeta gave to me.

Contents

1 Introduction

The thesis was conducted within ITS lab in an industrial project in collaboration with Kappazeta (project "Noise Modelling for Refining Farming Activities Recognition Based on Sentinel Data").

1.1 General View

Farming events detection is a task from an agricultural area aiming to solve the vegetation changes activity on fields or their nonappearance.

There are several reasons for monitoring motion on fields, but the most relevant for the European region is the subsidies program. Europeans common agricultural policy (CAP) provides income support for farmers. The subsidies program implies environmentally sustainable farming. Regarding the fields, it includes regularity of mowing, growing only specific crops, etc. To check whether farmers fail to obligate conditions, inspectors accomplish field visits. Depending on the country the number of involved inspectors may vary from hundreds to thousands. With the help of satellite data their work can be partially automated. It can allow to save the workforce costs and also free people for higher valued jobs.

Sentinel-1 and -2 are the Earth observation satellites held as projects of European Space Agency Copernicus missions. Sentinel-1 transmits data from synthetic-aperture radar (SAR) gained in the microwave range. Sentinel-2 carries a multi-spectral instrument (MSI) which works in visible, near and short infrared spectrum. Satellite imagery data is commonly used for management in agricultural, marine, accident domains. It is regular, reliable and near real-time accessible. For agricultural area, S-1 and S-2 time series are highly applicable, because they have a short guaranteed revisit interval (5 days for S-2, 6 days for S-1). It allows to track the changes in fields with an accuracy of several days. The spacial resolution of satellite imagery allows to follow changes of vegetation on each field and every land parcel.

Deep learning became a popular tool for both classification and regression tasks. In particular it is applicable for analysis of time series. Deep learning methods, given a large amount of data, outperform classical machine learning techniques. Even with a wide variety and continuous invention of new techniques there is no universal recipe for an any-purpose model. The involvement of deep neural networks still requires an individual approach at each task.

DeepSense is a deep learning method developed especially for time series analysis. Its input determined in terms of sensor - a source of regular measurements. The goal of this thesis is to adapt the DeepSense approach for farming activity detection, in particular on moving. Moving (cutting) is the most relevant activity for the subsidy program in Estonia and the most simplistic in contrast to tasks of crop classification. A neural network with a DeepSense architecture approach was developed. Experiments,

implementation changes and evaluation was based on the data which were pre-processed by Kappazeta. Kappazeta OÜ is an Estonian company that is working in the agricultural domain developing Earth observation applications and services.

1.2 Road map

In the second chapter there is a literature review of related work. Sentinel imagery became widely used for the remote sensing and, particularly, for control and observation of vegetation.

In the third chapter an introduction of data and developed model goes. A description of data sources, their flaws and assets. A description of data preparation and proposed model architecture.

In the fourth chapter experiments are described which were held to improve the baseline model performance. Comparisons of different hyper-parameters and their contribution to the result improvement.

In the fifth chapter conclusion is done describing the obtained results. Also, future study is discussed.

2 Related Work

After the launch of the Copernicus program the topic of events detection over Earth surface became commonly spread in the research community. Usage of satellite imagery allowed to achieve results in monitoring of urban areas, marine regions, emergency events and agricultural fields.

There is a narrow range of unique features that can be retrieved from satellite imagery and which are relevant for agricultural events. One of the first introduced is a normalized difference vegetation index (NDVI) [2]. This feature can be obtained from Sentinel-2 optical data. It is a well-established indicator for monitoring vegetation over a captured area.

Another type of feature can be obtained from synthetic-aperture radar (SAR) data. Such an instrument is carried by Sentinel-1 satellite mission.

SAR interferometric data were examined in a study[7] to be meaningful in terms of vegetation biomass changes. An extensive campaign covering 11 agricultural grasslands were held to collect reference data. Inverted correlations were found between vegetation height and temporal coherence calculated from consecutive SAR images of those fields. Grass removal was found to increase the level of coherence. It indicates, according to the authors, the potential of this parameter to be used for the detection of mowing. Authors also mention that precipitation appeared between the consecutive SAR measurements can decrease the discovered effect.

Features obtained from SAR repeat pass coherence are relatively new. VV (vertical transmit, vertical receive polarization) and VH (vertical transmit, horizontal receive polarization). The measure of median coherence was shown to be statistically significant over those features for moving events detection [8]. Mowing events recorded on agricultural grasslands in Central Estonia over the vegetative season of 2015 was analyzed. In this work authors were calculating coherence between each 12-day pairs of Sentinel-1 SAR images. Also the same calculations were applied for the 6-day interval images combining Sentinel-1A and -1B data.

A correlation analysis performed in [11] was directed on SAR images of Sentinel-1 dual-polarized (VV and VH) bands with vegetation and soil conditions. The explored images cover the South Tyrol mountain region (Italy) with time boundaries from October 2014 to September 2016. Authors have shown that backscattering coefficients acquired from S-1 SAR dual-polarized VH signal have a high level of correlation with the Normalized Difference Vegetation Index. Collected results allowed to detect phenological cycles in different vegetation cover types and phenological phases in meadows areas, with an accuracy compatible with the temporal resolution to S-1. Authors reported S-1 SAR data as a robust source of mowing period detection and less effective for start of season detection. Also results showed that with increasing altitude the accuracy drops down and acceptable under 1500 m a.s.l.

An algorithm for harvesting completion detection was proposed in [12]. The authors

were using Sentinel-1 SAR imagery. For the study the backscattering coefficient was calculated from VH and coherence was derived from VV polarization. Tests were held over the north of Kazakhstan during the season of 2018. Results in determining the harvesting completion dates got mean absolute error $= 6.5$ days. Those are comparable with the frequency of region revisiting of exploited by authors Sentinel-1B of 12 days. Authors claimed that adding Sentinel-1A imagery should improve their method because the revisiting frequency will increase.

In another study [13], Sentinel-1 SAR imagery coherence was compared to NDVI and ground truth data collected from land parcels. The experiment was held for the Flevopolder region, the Netherlands during a season in 2017. Five key crops were monitored: sugar beet, potato, maize, wheat, and English ryegrass. The authors observed the increase of coherence after harvesting. Moreover, they reported that coherence is a useful indicator of mowing events on the agricultural lands. Also, taking into consideration the high temporal revisit for the Netherlands, S-1 data has a significant potential to allow monitoring of growth and development of crops.

Multilayer perceptron neural networks were applied in study [14] for the detection of moving events. The authors used backscattering coefficients obtained from Sentinael-1 SAR C-band VV and VH polarization imagery. Also second-order texture metrics (homogeneity, entropy, contrast and dissimilarity) were used in a proposed method. The data was collected for the state Bavaria, Germany and explored the season of 2016. The authors reported that their approach was able to perform with overall accuracy of 85.71% for the validation set.

Another type of agricultural applications for satellite imagery is a crop classification.

It was shown that with the help of Landsat-8 (optical spectrum) and Sentinel-1 data major crops (wheat, maize, sunflower, soybeans, and sugar beet) in Ukraine were predicted with more than 85% accuracy [6]. The study was held for the Kyiv region, Ukraine with data obtained during the season of 2015. Authors were using CNN based models for the predictions. One-dimensional (1-D) CNN for spectral domain and two-dimensional (2-D) CNN for spatial (pixel-based) domain.

A method introduced in [10] incorporates Sentinel-1 and -2 data imagery to detect whether a parcel has a crop on it and its type. Covering more than 15,000 parcels, authors claimed to reach 90% accuracy classifying the crop types. Also the authors contributed the street-level imagery for improving their results. Deep learning was incorporated into a proposed framework. The used data was unbalanced in classes, thus authors had limited possibilities to gain high accuracy for the detection of narrowly represented crops. However, it is reported a 98% precision for detecting parcels which were wrongly declared as grasslands.

Described above NDVI, VV and VH coherence features were used in this thesis.

DeepSense approach was shown to be effective at time series analysis tasks [1]. The authors method significantly outperformed the state-of-art techniques in car tracking

with motion sensors, heterogeneous human activity recognition, user identification with biometric motion analysis. The technique incorporates both convolutional (CNN) and recurrent(RNN) neural network layers. Also authors claim that their method is a powerful noise reduction model. That property is highly applicable for time series derived from satellite imagery. In particular, NDVI relies on optical data which is exposed to some corruption with clouds. The coherence of VV and VH are affected by noise originating from the weather.

3 Methodology and contribution

This section describes the input data characteristics. Data is collected from satellite imagery and pre-processed by Kappazeta. Also the general architecture of the DeepSense model is presented. All processes are performed into the virtual environment of Python3 a popular programming language for scientific purposes. For a neural network setups Keras library is used with Tensorflow backend.

3.1 Dataset description

Data was used for Estonia of 2018 and 2017 observation years. Dataset has about 1700 fields (parcels) for 2018 and more than 1300 fields for 2017. The feature set originates from Sentinel-1 and -2 satellite imagery. Those satellites are a part of the Copernicus mission. Images are provided freely as an open Earth observation data. The data set used for this thesis is derived from raw images, but the process of its preparation will not be described here. Data processing was performed and the resulting dataset was provided by Kappazeta OÜ.

Three main features are present in data: NDVI obtained from Sentinel-2; coherence VV and VH obtained from Sentinel-1. In Figure 1 NDVI, CohVV and CohVH are displayed. Those features take values from 0 to 1. The average period between consecutive measurements is 6 days for NDVI and 3 days for coherence.

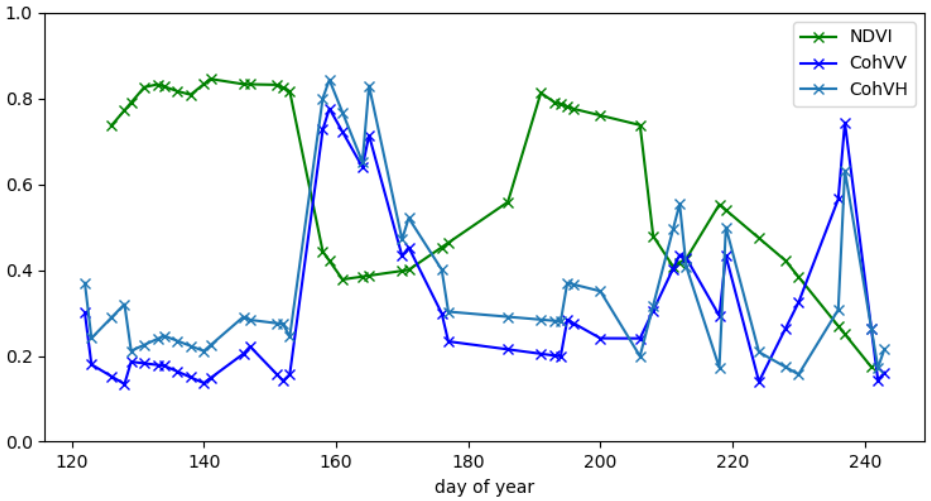


Figure 1. Example of NDVI, CohVV and CohVH features for a parcel during a season

3.1.1 Sentinel-1 feature set

Sentinel-1 is a satellite mission that aims to deliver synthetic aperture radar (SAR) images of the surface of Earth. SAR images are done with a decent spacial resolution. Moreover because the radar works in a microwave band, produced images collection does not interfere with the cloud conditions. Sentinel-1 imagery has VV (vertical polarization) ans VH (horizontal polarization) measurement channels. The coherence of both VV ans VH was used because it was shown in [8] that all of these features are sensitive to agricultural events, particularly to changes in vegetation. Coherence is a measure that reflects the similarity between a SAR image and the successive one.

$$\gamma = \frac{|\langle s_1 s_2^* \rangle|}{\sqrt{\langle s_1 s_1^* \rangle \langle s_2 s_2^* \rangle}} \qquad (1)$$

In (1) $s1$ and $s2$ denote two complex synthetic aperture radar (SAR) acquisitions, s^* denotes the complex conjugate of s, $|\cdot|$ denotes the absolute value, and $\langle \cdot \rangle$ denotes the spatial average over a window of a certain size.

The images must have the same relative orbit number so they were made at the same angle. If the mowing event happens during the period that is between of two images were captured, the coherence will increase temporarily. After the event it will remain at a relatively high level because a small amount of vegetation does not produce a lot of mismatching on the images. Not mowed grass is affected by weather conditions like wind and rain. It is highly capable of producing dissimilarities on the consecutive SAR images. At the same time mowed grassland remains more similar until the vegetation gets recovered. Thus, the signature of the event occurring is a temporal increase of VV and VH coherence values.

Numerically VH and VV coherence is normalized and takes values between 0 and 1. Measurements are regular with at most a 6-day interval. The absence of values on implied days does not occur. Values may have noise. The noise is introduced by several circumstances like weather (wind and rain), and a changing amount of moisture in some types of fields (river valleys, grasslands, wetlands, etc.)

3.1.2 Sentinel-2 feature set

Sentinel-2 is another Earth observation mission held by the European Space Agency (ESA). The satellite cares an optical device (multi-spectral instrument) and provides imagery in the visible, near and short infrared spectrum.

The main difference with the Sentinel-1 features is data regularity. As images are captured in an optical range those are affected by weather, clouds in particular. ESA provides Sentinel-2 data together with a cloud mask. The maintained mask supports a reasonably good accuracy over the image. Most of an error is contributed by small clouds.

Normalized difference vegetation index (NDVI) is the feature which is obtained from Sentinel-2 imagery.

$$\text{NDVI} = \frac{\text{NIR} - \text{RED}}{\text{NIR} + \text{RED}} \tag{2}$$

In (2) NIR stands for the near-infrared and RED stands for the red spectrum correspondingly. This characteristic originates from the properties of green plants. In greater extent those reflect light in near-infrared band and mostly absorb in the red spectrum. NDVI is a well-established and commonly used indicator for vegetation in the agricultural area which is based on optical data. The signature of the event in terms of NDVI feature is a fast decrease and gradual growth after it.

Numerically NDVI is also normalized and takes values between 0 and 1. Measurements are affected by irregularity due to the cloud coverage. Approximately 75% of observations during a season over Estonia are discharged because of weather conditions. The cloud mask provided with the imagery does not have ideal accuracy. Sometimes fields can be covered partially by a cloud which leads to an outlying observations and increases a variance of the data. Due to the spatial resolution of data and sometimes relatively small field size if the event occurred only on the part of a filed it will not be reflected in time series.

3.1.3 Analysis of Frequency for Sentinel-1 and -2 data points

According to ESA description Sentinel-1 has a guaranteed exact repeat cycle of 6 days. For Sentinel-2 this repeat cycle is equal to 5 days. In fact Sentinel-1 and -2 each work as a constellation of two satellites: S-1A and S-1B, S-2A and S-2B. It allows to double the revisit frequency. For example, from 12 to 6 days for S-1. Because the Estonian region is placed at larger latitudes, the time difference between satellite imagery acquisitions may be even lower.

On Figure 1 a box plot of the amount of days between data points for S-1 features (CohVV and CohVH) and for S-2 feature (NDVI). It can be seen that NDVI frequency has a lot of outlying values. It happens because of cloud coverage: not all S-2 images capture exactly the land of a field. For S-1 frequencies most of the values remain between 1 and 6 days.

In the Table 1 it can be seen that S-1 data acquisition frequency has a much lower variance than S-2 data.

	Sentinel-1	Sentinel-2
mean	3.407	6.140
std	2.386	5.952

Table 1. Mean and standard deviation of revisit frequencies for Sentinel-1 (CohVV, CohVH) and Sentinel-2 (NDVI) data in days

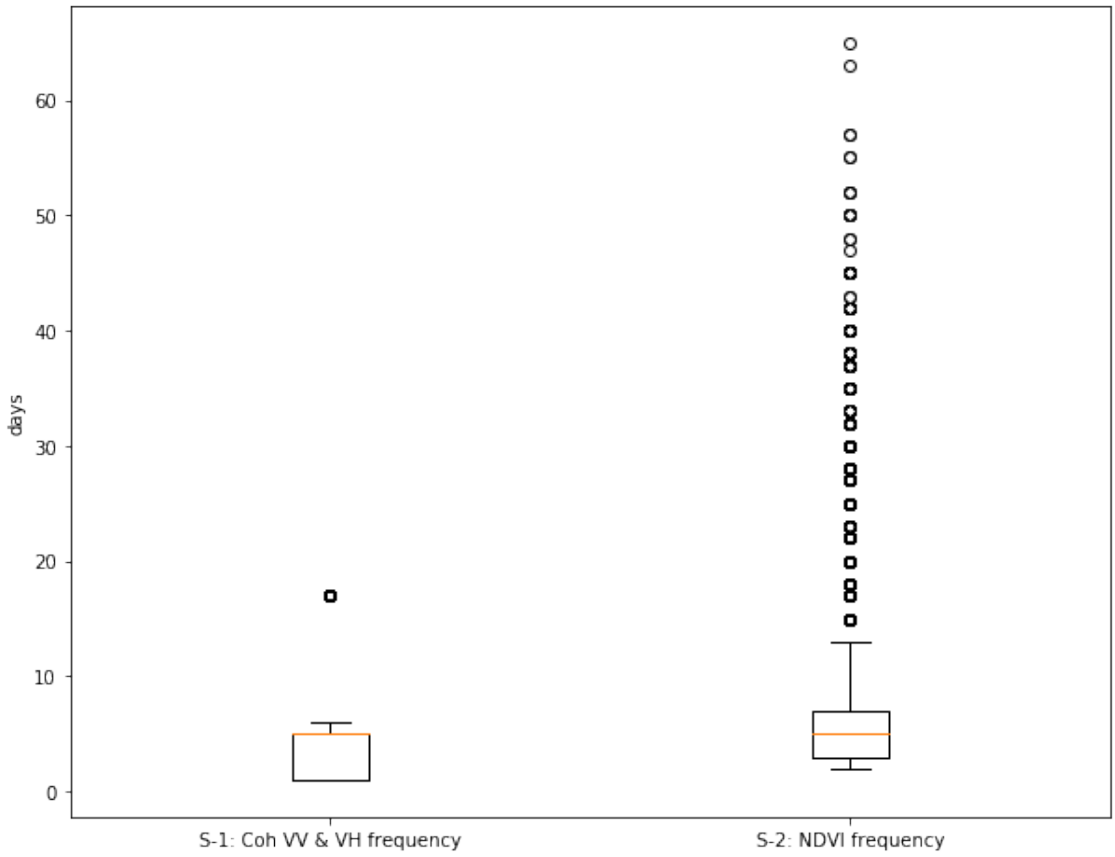


Figure 2. Box plot diagram describing frequency of revisit for Sentinel-1 (CohVV, CohVH) and Sentinel-2 (NDVI) data in days.

3.1.4 NDVI outlier

During data analysis it was found that the NDVI feature set struggles with a specific type of outliers. A rapid fall from a relatively high value with a consecutive return to the values of the previous magnitude. To check the fraction of such anomalies a detection method was implemented.

For time points $t_1, t_2, t_3, ...$ which have an NDVI value for an arbitrary parcel we calculate the gradient (slope):

$$slope(t_i) = \frac{NDVI(t_i) - NDVI(t_{i-1})}{t_i - t_{i-1}} \tag{3}$$

In (3) t is calculated in days.

Then we calculated the difference of the slopes divided by the time difference:

$$diff(t_i) = \frac{slope(t_i) - slope(t_{i-1})}{t_i - t_{i-2}} \tag{4}$$

Basically from equation (4) it can be seen that calculated $diff$ is nothing else than a numeric second derivative.

As the rapid decrease of the NDVI magnitude is usually a signature of a moving event, the biggest interest cause by those data points which are showing a dramatic decrease of magnitude. The points which correspond to such behavior have a positive $diff$ calculated. If we define a threshold for a $diff$

$$th = mean(DIFF) + 3 * std(DIFF) \tag{5}$$

The data points which exceed a threshold $diff > th$ will take a 1% fraction of the whole dataset.

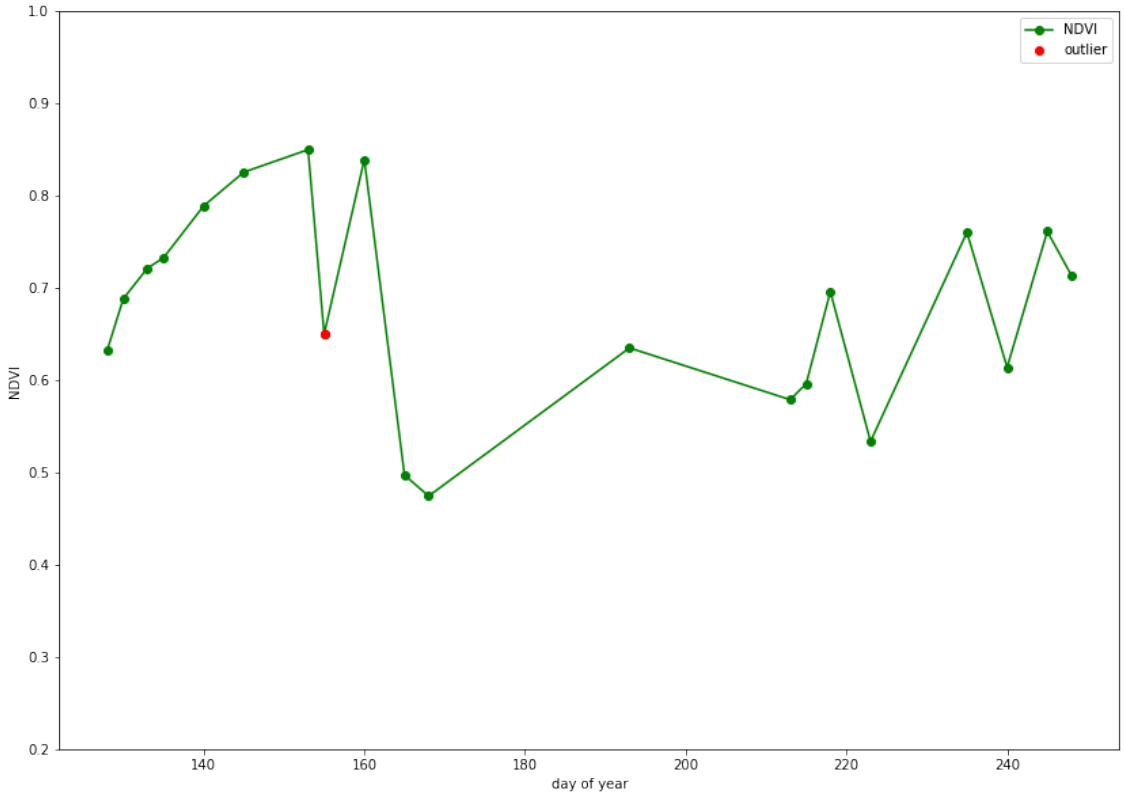


Figure 3. Example of outlying data point for NDVI series. The laydown in day 155 is to sharp and exceeds the threshold of the three standard deviations.

An example of outlier can be seen in Figure 3. Such kind of outliers occurs due to the optical nature of the NDVI measurement. As the cloud mask coverage does not always work perfectly there are possible cases when a parcel was covered by a cloud partially or exposed to a shadow and NDVI measurement was added to the data set. That leads to such anomalies in data.

3.1.5 Labeling

Labels were obtained from field books. Farmers keep the records with dates of their farming activities. Kappazeta has an agreement with farmers to provide those field books. For 2018 there is information about more than 1700 parcels collected.

It was mentioned by Kappazeta that there are time shifts between the event signatures and marks that were left by farmers. The difference could reach up to 7 days.

3.2 Model architecture

Deep Sense showed itself as a powerful noise reduction mechanism. It was shown to be beating state-of-art algorithms for time series analysis [1]. The DeepSense approach aggregates two types of neural network techniques that are used for time series analysis: convolutional and recurrent neural networks. Such a joint usage can allow exploiting positive properties of each type. Convolutional neural networks can be resistant to noise and outliers in time series, recursive ones are good at finding dependencies along the whole series observation period.

The architecture of the neural network with DeepSense technique consists of 4 logical parts:

- Individual sensor sub-network,
- Merging sensors sub-network,
- Recursive sub-network,
- Final prediction sub-network with attention mechanism.

In the Figure 4 a variant of the DeepSense implementation which was used in this thesis. The current architecture consists of two convolutional layers, 2 flatten and concatenation layers, two stacked together recurrent layers and an output.

3.2.1 Input preparation

Before entering the model data has to be prepared. There is a defined mechanism of how to do that which is described below.

The model could have k sensors. Each sensor's data may have d dimensions in space and t dimensions in time (since it is a time series). Each sensory input of size $d \times t$ is split into not overlapping sub-sequences (windows) of size $d \times T$. Thus, $t = n \cdot T$ where n is an amount of those non-overlapping intervals. Afterwards the sensory input tensor X will have size $T \times n \times d$. The total input of k sensors will look like $\mathrm{X} = \{X_1, X_2, X_3, ..., X_k\}$.

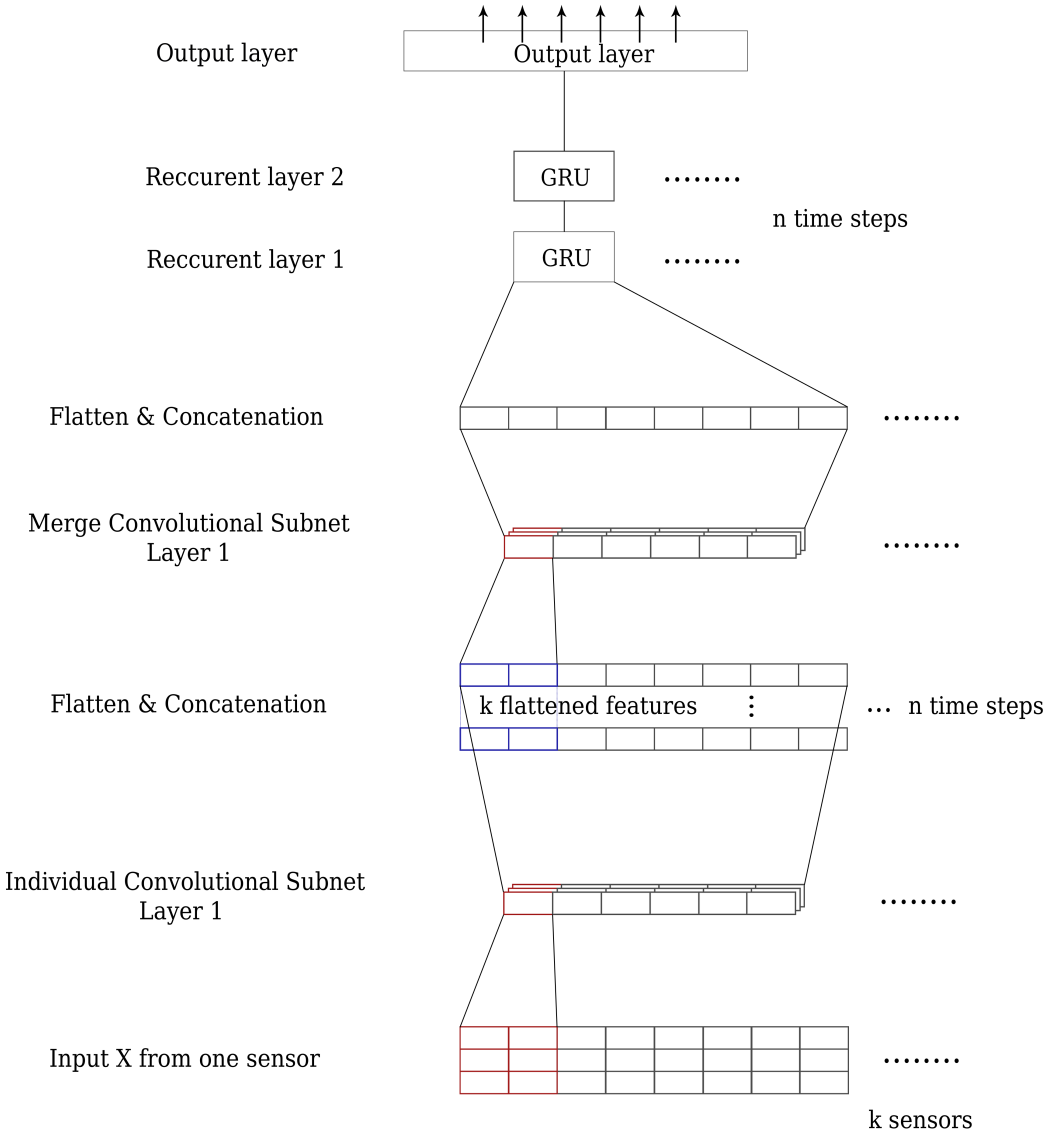


Figure 4. Variant of DeepSense architecture implementation. Individual sensor sub-network contains one convolutional layer. Merging sensors sub-network contains one convolutional layer. In total 2 convolutional layers, 2 flatten and concatenation, 2 stacked recurrent layers, one output layer

3.2.2 Individual sub-network

The Individual sub-network consists of a two-dimensional convolutional layer. The kernel of size $[\_, d]$ is applied. _ means that at this dimension the kernel can be adjusted and d still stands for the amount of dimensions of a sensor. There can be several convolutional layers each followed by a batch normalization layer. In the original paper there were 3 convolutional layers used. Subsequently the output is reshaped to be flattened.

For our purposes we will use only one convolutional layer with kernel [2, 10], stride [1, 1], padding valid. That layer will be followed by a batch normalization with momentum $= 0.99$.

3.2.3 Merging sub-network

The Merging sub-network is also convolutional. It is supposed to reveal the dependencies among sensors. Each individual sensor sub-network output is collected and concatenated together in a new dimension. Afterward the similar convolutional layers with subsequent batch normalization are applied. The kernel size in this case is equal to $[\_, k]$, where k stands for the number of sensors entering the model. _ means that the parameter is adjustable.

In the original paper authors again used 3 convolutional layers. For our case we limit the amount of layers to one. It has kernel of size [2, 30] and stride [2, 2], padding valid. The same as for the Individual sub-network, convolutional layer is followed by a batch normalization with momentum $= 0.99$.

3.2.4 Recursive sub-network

The Individual and Merged sub-networks are applied for each window and then enter the recursive sub-network.

Recursive sub-network adopts GRU (grated recursive unit) components. To increase the capacity of the model the sub-network consists of two stacked GRU layers. There is a dropout layer between them for regularization purposes.

From available recursive network architectures we chose GRU component as it was done in the original paper. GRU was proven to have a comparable performance with LSTM (long short term memory) but with lower computational power needs [16]. In our case we used two stacked GRU components each of size 100 with a dropout layer between of level 0.5 .

3.2.5 Attention layer

Attention mechanism is a technique which became popular due to an impact on machine translation models [17]. The main principle it uses is a weighted sum of the input values,

but the weights are trainable. We used an approach [15] called general attention described in (6).

$$\begin{aligned} score &= h_s^T W h_t \\ attention\_weights &= Softmax(score) \\ context\_vector &= h_s \cdot attention\_weights \\ attention\_output &= [context\_vector; h_t] \end{aligned} \tag{6}$$

The states h_s of a Recursive layer enter the attention sub-network. Those are multiplied by a matrix of weights W. And then multiplied with a dot product by the last hidden state $h_t$from the GRU output. Attention weights are calculated as a soft-max from the score. A context vector is a multiplication of attention weights and hidden states. The attention vector is a dense layer from the concatenation of context vector and the last hidden state.

$$Softmax(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{7}$$

Softmax function (7) transforms the vector of score into probability-like scores which in sum gives 1.

The final output of an attention mechanism is passed to a fully connected layer of size 500 without bias and with activation function tanh.

3.2.6 Final layer

Final prediction sub-network adopts the last fully connected with sigmoid (8) activation to predict events. The value range of this function lays between 0 and 1, which is appropriate for our binary classification task.

$$sigmoid(x) = \frac{1}{1 - e^{-x}} \tag{8}$$

Thus, the final output of the model is a vector of mowing event probabilities for each day of an observations period. We chose the period from 15th April to 11th September . In total that period covers 150 days.

3.2.7 Applied architecture

The architecture aggregates a convolutional neural network technique. The data points still struggle from big gaps of the size of revisit period for CohVV, CohVH and even longer gaps for NDVI data. To improve that linear interpolation is applied.

In the Figure 5 can be seen that there exists a data point for each observed day. Those values are obtained from interpolation of available data points.

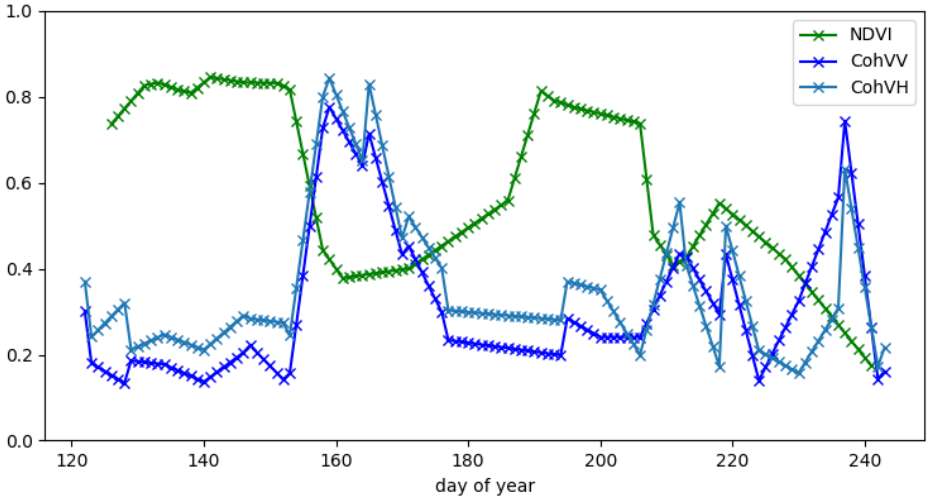


Figure 5. Example of NDVI, CohVV and CohVH features for a parcel during a season

The DeepSense approach implies the presence of sensors, the input data has to be logically separated into several sensory inputs. In this work we propose to combine CohVV and CohVH together as a two-dimensional input. NDVI features are entering the network as a one-dimensional input. The input series for each sensory input are split into a set of T not overlapping sequences.

We implement a model with 2 sensors for input NDVI and coherence (VV and VH). Observation period of season was split into windows of 10 days.

3.3 Purely convolutional model

Kappazeta already had a flagship model based on a CNN. We re-implemented the architecture by the given description. As an input the model takes a tensor with NDVI, CohVV and CohVH features for each day of a season for each parcel. There are missing values in series, because the revisit period of satellites is longer than one day. For NDVI (Sentinel-2) features it supposed to be not longer than 5 days, but a big amount of data points are missing due to the cloud coverage. For CohVV and CohVH (Sentinel-1) features it is not longer than 6 days and not affected by clouds. Missing values are interpolated linearly. The example of interpolation can be seen in Figure 5. The model consists of the next layers.

1. 1-d convolutional layer:
   20 filters, kernel size 20, same padding, activation function ReLU

2. Batch normalization:
   momentum 0.99, without scaling

3. 1-d convolutional layer:
   20 filters, kernel size 20, same padding, activation function ReLU

4. Batch normalization:
   momentum 0.99, without scaling

5. Final convolutional layer with 1 filter and final activation function sigmoid

The output of the model is a vector of event probabilities for each day of a season.

3.4 Training Loss

The task lies in a field of classification problem. There are two classes to be predicted: positive and negative events. Binary cross-entropy is the appropriate metric for a loss function.

$$L_{BCE} = \frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{9}$$

In the equation (9) binary cross-entropy loss is shown. y_i stands for a real true value, $p(y_i)$ is a probability of a class predicted by a model.

The issue is that input data has an imbalanced distribution of events. There are more negative (zeros) than positive events (ones). To make model more sensitive for positive events a weighted binary cross-entropy was used.

$$L_{weighted\text{-}BCE} = \frac{1}{N}\sum_{i=1}^{N} w_1 \cdot y_i \cdot log(p(y_i)) + w_0 \cdot (1 - y_i) \cdot log(1 - p(y_i)) \tag{10}$$

In the formula (10) w_1 and w_0 stand for a weights for a corresponding classes: 1 and 0. The idea of such weights is to make our model paying more attention to less represented classes during training process.

$$\begin{aligned} w_0 &= \frac{I + O}{O} \\ w_1 &= \frac{I + O}{I} \end{aligned} \tag{11}$$

In (11) I means the total amount of positive events in a training set (ones) and O means the total amount of negative events (zeros).

3.5 Summary

In this chapter data was described with a methodology of its preparation. The input features originate from Sentinel-1 and -2 imagery. NDVI is obtained from S-2 imagery. It suffers from cloud coverage. Thus there are a lot of missing data points in the data set. Also the NDVI feature suffers from outliers that appear due to the inaccuracy of a cloud mask coverage. The coherence of VV and VH are obtained from S-1 imagery. S-1 SAR equipment is not affected by cloud coverage. Thus the features have a good regularity which corresponds to a revisit frequency for S-1.

DeepSense framework architecture is characterized by its main components (sub-networks): Individual sensor sub-network, Merging sensors sub-network, Recursive sub-network, Final prediction sub-network with an attention mechanism. For each sub-network that adopts convolutions only one layer was used.

Previously constructed by Kappazeta mowing events prediction neural network uses only convolutional layers.

The training loss function was described. Weighted binary cross-entropy loss is chosen for two reasons: classification task implies only two classes (positive and negative events), classes have an imbalanced representation in the dataset.

4 Results and analysis

In this section the training and eveluation of the models is discussed. The DeepSense approach and the purely convolutional network are compared in terms of the event accuracy and the end of season accuracy. The DeepSense model is tested with different architectural changes to get a better performance.

4.1 Experiments design

To test the properties of the DeepSense and the purely convolutional model a set of experiments was held. To evaluate the performance of the models two metrics were used. The event accuracy metric was used to represent the quality of the models in terms of predictions for each day. The end of season accuracy was used to represent the quality of the models in terms of the whole season prediction.

In this analysis, we used two different years 2017 and 2018. The year 2017 has the characteristic of being less dense with respect to the amount of parcels; however, the year 2018 has more parcels and denser by 30%. In addition, according to the Estonian Weather Service comparing to the normal period (from 1981 to 2010): 2017 had a cooler spring 4,5ºC (normal 4,6°C) with a bit lower precipitation total than normal 104 mm (normal 110mm), summer was cooler 15,2ºC (anomaly –0,8°C) and with lower precipitation 204 mm (91% of normal). 2018 had a spring warmer than normal 5,6 °C (normal 4,6 °C) and with precipitation lower than normal 78 mm (normal 110 mm), summer was very warm 17,8 °C (anomaly 1,8 °C) and with lower precipitation total 149 mm (67% of normal). Overall it can be concluded that 2018 was unusually warmer and with a lower precipitation in contrast to the year 2017 and other years before.

The training process was held. Both models (purely convolutional and the DeepSense) were trained and tested for the years 2017 and 2018. The 5-fold cross-validation was held for the data of years 2017 and 2018 separately. The models were trained on the data for the year 2017 and tested on the data of the year 2018. Also both architectures (DeepSense and CNN) were tested with the 5-fold cross-validation on the merged data from the years 2017 and 2018.

Tuning experiments were done with the DeepSense model. The model was tested with the help of 5-fold cross-validation on the data from 2018. 2018 data is preferred over 2017 because it contains a substantially bigger number of observed parcels. For the tuning experiments the data only from one year was chosen to save the weather conditions consistency. Different aspects of the DeepSense architecture were explored including experiments with all its sub-networks: Individual, Merged, Recurrent and Output.

4.2 Evaluation metrics

To evaluate a quality of a model several metrics were used. Event accuracy and end of season (EOS) accuracy were the main to assess the performance. To define those metrics TP (true positive), TN (true negative), FP (false positive), FN (false negative) have to be described.

- TP (true positive) is the number of events that were predicted by model as positive and by the ground truth are positive
- TN (true negative) is the number of events that were predicted by model as negative and by the ground truth are negative
- FP (false positive) is the number of events that were predicted by model as positive but by the ground truth are negative
- FN (false negative) is the number of events that were predicted by model as negative but by the ground truth are positive.

Accuracy can be seen in (12).

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Also End Of Season (EOS) Accuracy was used. It can be seen in formula (13). EOS accuracy changes the level of observation from event to a season and pays attention to whether a positive event occurred during a season. If a positive event happened during the season at least once, the season is considered as a one with a positive event. If no positive events happened during season, the season is considered as one that has a negative event. With the same logic TP, TN, FP, FN are calculated but only on the level of seasons instead of events. Thus, an end of the season (EOS) accuracy is introduced.

$$ACCURACY_{EOS} = \frac{TP_{EOS} + TN_{EOS}}{TP_{EOS} + TN_{EOS} + FP_{EOS} + FN_{EOS}} \tag{13}$$

This EOS accuracy is used as the main metric to tune the model.

4.3 Training process

The models were trained for 100 epochs, with batch size 20, optimizer Nadam. Shuffle of data was turned on, which means that on each epoch parcels were entering the model in a different order. End of season accuracy for validation set was the monitored metric.

The models were tested with the method of K-fold cross-validation. Such a technique is applicable when the amount of data is limited. There are parcels in the data set which

does not have any grassland event on them. Meaning that there was no positive event during a season. As EOS accuracy is a metric that is supposed to represent the quality of a model, the fraction of fields with events and without events should remain the same in each fold. For that purpose a StaticKFold method from Scikit-learn library was applied. Amount of folds $= 5$.

4.3.1 Training and testing over the year 2018

From the Tables 2 and 3 it can be seen that purely **convolutional** model gains much better event accuracy than a proposed DeepSense architecture: 96% against 89%. At the same time both neural models show almost the same performance at EOS accuracy reaching slightly more than 90%.

	Event acc	EOS acc
avr	89.41%	90.66%
std	2.04%	0.95%

Table 2. Event accuracy and end of season (EOS) accuracy obtained from 5-fold cross-validation of year 2018. **DeepSense architecture**

	Event acc	EOS acc
avr	96.55%	90.56%
std	0.64%	1.72%

Table 3. Event accuracy and end of season (EOS) accuracy obtained from 5-fold cross-validation of year 2018. Purely **convolutional architecture**

As the quality of a proposed model according to the tracked metrics was not exceeding the performance of the previous purely convolutional model, further improvements of the implemented DeepSense architecture were required.

From Figure 6 changes of training and validation loss can be discovered for a purely convolutional model. The model was not improving its validation loss after approximately 20 epochs. On Figure 7 event accuracy (a) and end of season accuracy (b) can be seen for the same model. According to the accuracy graphs the purely convolutional model quickly gains a good level of the event accuracy, but remains almost constant for the EOS metric. The presence of at least one positive event marks the season for a parcel as positive. According to the EOS accuracy training history model fails not to mark positive events for those parcels which did not have one.

In Figure 8 the history of training and validation loss can be seen for the DeepSense architecture. In Figure 9 the training history of event accuracy and end of season accuracy are displayed. The model is able to improve on the training set after each epoch meaning

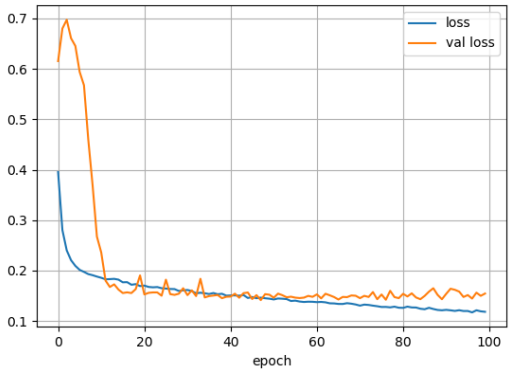
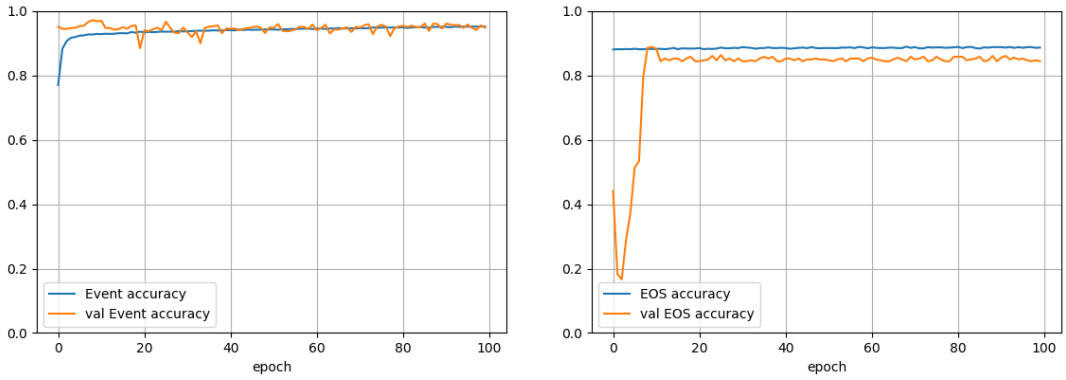


Figure 6. Training history of the purely **convolutional model**. Validation and training loss graphs for one fold. Year 2018.



(a) Event accuracy

(b) End of Season Accuracy

Figure 7. Training history of the purely **convolutional model**. Event and end of season accuracy for one fold for validation and training sets. Year 2018.

that it has enough capacity to generalize the data. At the same time validation loss starts growing. The issue of growing validation loss may be caused by several reasons. Used weighted binary cross-entropy makes wrongly predicted positive events to contribute more. Cross-entropy is not bounded loss, thus some outliers in the validation set may cause such behavior contributing in loss more during each new epoch.

For both purely convolutional and the DeepSense models EOS accuracy reaches the point and do not improve during training. For 2018 data the fraction of fields with events in them approximately equals to 88%. Both CNN and the DeepSense models hit the plateau during training at that value for the end of season accuracy. It means that the

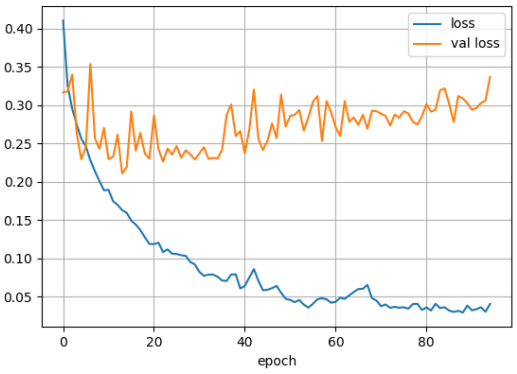
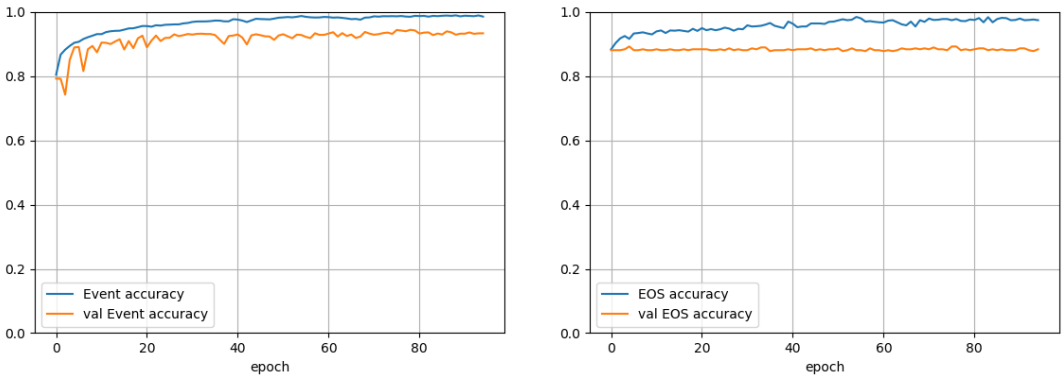


Figure 8. Training history of the **DeepSense model**. Validation and training loss graphs for one fold. Year 2018.



(a) Event accuracy

(b) End of Season Accuracy

Figure 9. Training history of the **DeepSense model**. Event and end of season accuracy for one fold for validation and training sets. Year 2018.

models struggle to predict correctly those fields that did not have any positive event in them.

4.3.2 Training and testing over the year 2017

The cross-validation process was also held for the data obtained from 2017.

	Event acc	EOS acc
avr	91.81%	88.63%
std	1.14%	3.13%

Table 4. Event accuracy and end of season (EOS) accuracy obtained from 5-fold cross-validation of year 2017. **DeepSense architecture**

	Event acc	EOS acc
avr	92.16%	96.30%
std	7.96%	1.07%

Table 5. Event accuracy and end of season (EOS) accuracy obtained from 5-fold cross-validation of year 2017. Purely **convolutional architecture**

In terms of event accuracy the DeepSense model reached similar to the purely convolutional network on average of 92%. At the same time the variance of the obtained DeepSense model event accuracy was lower meaning that it gains a much more stable performance.

The DeepSense architecture experienced a worse performance in EOS accuracy on the data from year 2017 than the purely convolutional model. While the CNN model reached more than 96% EOS accuracy in average for 5-fold cross-validation, DeepSense approach gained only more then 88%.

4.3.3 Training on 2017 and testing on 2018

To validate the ability of the developed DeepSense model to be trained on a transfer data the models were trained on 2017 data and tested on 2018 for the DeepSense in Figure 6 and for the purely convolutional network in Figure 7.

	Event acc	EOS acc
value	85.94%	88.86%

Table 6. Event accuracy and end of season (EOS) accuracy. Trained on 2017 tested on 2018. **DeepSense architecture**

	Event acc	EOS acc
value	92.12%	92.41%

Table 7. Event accuracy and end of season (EOS) accuracy. Trained on 2017 tested on 2018. Purely **convolutional architecture**

It can be seen that the purely convolutional model (Table 7) gained much better results than the DeepSense implementation (Table 6). Data from 2017 was not generalized well by DeepSense model, thus accuracy for year 2018 remained shallow.

There are two possible reasons of the lower performance gained the DeepSense model that was trained with the data from 2017 and tested on the year 2018. At first, the accuracy of labeling done by farmers can vary from year to year. Secondly, the year 2018 had an anomalously warmer and drier farming season than 2017. Thus, the model struggled to adapt those changes.

4.3.4 Training and testing on the merged 2017 and 2018 data

The two datatsets from 2017 and 2018 were merged together. Both CNN and the DeepSense models were tested with the 5-fold cross-validation.

	Event acc	EOS acc
avr	90.67%	89.92%
std	3.12%	3.47%

Table 8. Event accuracy and end of season (EOS) accuracy obtained from 5-fold cross-validation of merged years 2017 and 2018. **DeepSense architecture**

	Event acc	EOS acc
avr	85.34%	90.32%
std	9.17%	2.07%

Table 9. Event accuracy and end of season (EOS) accuracy obtained from 5-fold cross-validation of merged years 2017 and 2018. Purely **convolutional architecture**

The DeepSense performance can be seen in Table 8 and the performance of convolutional architecture in Table 9. Both models gained almost the same EOS accuracy of 90%. The DeepSense architecture allowed to reach a much better event accuracy of 90% in contrast to the purely convolutional model with 85%.

Probably the DeepSense model was capable to grasp the changes of the weather conditions that reflected in a better event accuracy.

4.4 Tuning experiments

The DeepSense model has a relatively high capacity: its architecture has a lot more trainable parameters in contrast to the purely convolutional network. We choose the year 2018 because it has a substantially bigger number of observed fields than dataset from 2017 (1700 parcels against 1300). Also the datasets of 2018 and 2017 were not merged for this experiments to save the consistency of the weather conditions. Tweaks and experiments to improve the model characteristics are to be held. To find the best hyper-parameters of the model different modifications will be applied:

- Usage of attention mechanism
- Amount of convolutional layers into Individual and Merging sub-networks
- Kernels in convolutional layers
- Number of filters in convolutional layers
- Size of recurrent layers
- Dropout in recurrent sub-network
- Output sub-network middle layer size
- Activation functions for convolutional layers and output sub-network
- Optimizers

The model is tested with attention mechanism removed from the final output prediction sub-network. Instead the output of the Recurrent sub-network is concatenated and transmitted to the multi-layer perceptron containing two fully connected layers.

Individual an Merged sub-networks contain convolutional layers into them. The model is evaluated with different number of those layers into each sub-net. All the variants with one and two CNN layers are tested for each containing them sub-network.

After that another parameters of convolutional layers are explored. Different combinations of kernels were applied for the Merged and the Individual sub-networks. Also a number of filters into each CNN layer is tested.

After CNN based sub-networks parameters of the Recurrent were evaluated. Various combinations of the size of GRU layers are tested. Also the dropout parameter is checked for that sub-network.

The size of the middle layer in the output MLP is tested. Different combinations are examined: from the size of the output to several times bigger.

Also next, the activation functions in the convolutional layers are tested. Those are different options of linear unit: ReLU, leaky ReLU, ELU.

Finally the model is tested with other optimizers.

4.4.1 Final Layer without Attention

Attention for the DeepSence approach was proposed in the original paper [1] as a good option for time series prediction. Attention mechanism may not introduce improvement because of too strong reeducation of information from the recurrent output. Instead a simple two layer perceptron is used. GRU output is unstacked by the last axis and concatenated. That output is transmitted to a fully connected layer of size 500 and activation ReLU. Afterwards the output goes to the final fully connected layer of with activation sigmoid.

	Event acc	EOS acc
avr	94.08%	93.44%
std	1.49%	0.68%

Table 10. Test of DeepSense architecture with no attention mechanism. GRU output concatenated. Event accuracy and end of season (EOS) accuracy obtained from 5-fold cross-validation of year 2018.

The results can be seen in the Table 10. The model with DeepSense architecture gained a better results in both event accuracy and end of season (EOS) accuracy. EOS accuracy increased to over 93% outperforming a purely convolutional model.

4.4.2 Architecture experiments with the Individual and Merged sub-networks

The DeepSense model adopts two sub-networks that have a convolutional architecture: Individual and Merged.

Different amount of convolutional layers is used. In the baseline model each sub-network has only one such layer. We tested a different amount of CNN layers. It can be seen in Table 11. The baseline architecture with only one layer for each sub-network was not outperformed by other options with two layers in the Individual sub-net and with two layers in the Merged sub-net. The configuration with one layer in the Individual sub-network and two layers in the Merged sub-network gave almost the same performance for EOS accuracy and slightly better result for event accuracy. Due to insignificance the baseline option with one layer for each sub-net remained for further experiments.

For the second convolutional layer of Individual sub-network we used a filter with kernel [1, 3], stride [1, 1] and valid padding. For the second convolutional layer of Merged sub-network we used a filter with kernel [1, 30], stride [1, 1] and valid padding.

Individual sub-net layers	Merged sub-net layers		Event acc	EOS acc
1	1	avr	94.08%	93.44%
		std	1.49%	0.68%
1	2	avr	94.73%	93.44%
		std	1.05%	0.78%
2	1	avr	93.86%	92.86%
		std	1.29%	0.88%
2	2	avr	94.25%	93.15%
		std	1.08%	1.34%

Table 11. Different amount of convolutional layers for each sub-network performance. Measurements based on 5-fold cross-validation. Used metrics event accuracy, end of season (EOC) accuracy, year 2018

Another parameter in the convolutional layers that was tested is a kernel size. For the Individual sub-network kernel size should remain [2, _], because the coherence feature is used as a single 2-dimensional sensor. Also another parameter of kernel size can not exceed 10, because it equals to the size of window by which the input series were separated in sub-sequences. Stride was set to [1, 1]

For the Merged sub-network kernel size should also look like [2, _], but at this point because of the amount of sensors (NDVI and coherence). Stride was set to [2, 1].

Different size of kernels, were applied.

Individual sub-net kernel	Merged sub-net kernel		Event acc	EOS acc
[2, 5]	[2, 20]	avr	93.10%	93.27%
		std	1.07%	1.75%
[2, 10]	[2, 20]	avr	94.05%	92.75%
		std	1.36%	0.37%
[2, 5]	[2, 30]	avr	93.56%	92.86%
		std	1.50%	0.59%
[2, 10]	[2, 30]	avr	94.08%	93.44%
		std	1.49%	0.68%
[2, 5]	[2, 40]	avr	93.76%	92.51%
		std	1.87%	0.97%
[2, 10]	[2, 40]	avr	93.90%	92.92%
		std	1.86%	0.62%

Table 12. Performance of different kernels applied to individual and merged sub-networks layers. Based on 5-fold cross-validation. Used metrics event accuracy, end of season (EOC) accuracy, year 2018

In the Table 12 different kernels were applied for convolutional layers in the Individual and Merged sub-networks. The experiment did not find any improvements compared to the baseline configuration with kernels [2, 10] and [2, 30] corresponding to the Individual and Merged sub-nets.

Next parameter that was tested is amount of filters in the convolutional layers. Theoretically higher amount of filters may allow a model to learn more intermediate features. At the same time the increase of network parameters may cause over-fitting.

In Table 12 a results of applying different amount of filters is displayed. Configuration with 10 filters gained almost the same result as the baseline with 20 filters. Because the significant improvements were not found the baseline configuration remained.

Amount of filters		Event acc	EOS acc
10	avr	94.90%	93.27%
	std	0.94%	1.33%
20	avr	94.08%	93.44%
	std	1.49%	0.68%
40	avr	93.41%	92.74%
	std	1.86%	1.26%
60	avr	91.05%	91.47%
	std	2.44%	1.00%

Table 13. Performance of different amount of filters applied to individual and merged sub-networks layers. Based on 5-fold cross-validation. Used metrics event accuracy, end of season (EOC) accuracy, year 2018

4.4.3 Experiments with the Recurrent sub-network size

The next DeepSence architecture parameter that was checked the size of grated recursive units in the Recurrent sub-network. GRU components used hyperbolic tangent activation, and sigmoid as a recurrent activation.

In Table 14 different sizes of the first and second recurrent layers were tested. Explored combinations showed that the best performance remained for the baseline variant of configuration with 100 units for the first layer and 100 units for the second.

GRU1	GRU2		Event acc	EOS acc
100	50	avr	94.29%	93.61%
		std	1.30%	0.84%
100	100	avr	94.08%	93.44%
		std	1.49%	0.68%
200	50	avr	94.45%	94.08%
		std	1.36%	1.6%
200	100	avr	94.34%	93.67%
		std	1.21%	0.98%
200	200	avr	94.36%	93.62%
		std	1.45%	0.86%

Table 14. Performance of different amount of GRU units in the Recurrent sub-network. Based on 5-fold cross-validation. Used metrics event accuracy, end of season (EOC) accuracy, year 2018

4.4.4 Experiments with the Recurrent sub-network dropout

GRU dropout level was tested in the Recurrent sub-network.

Reccurent dropout		Event acc	EOS acc
0.1	avr	94.34%	92.22%
	std	1.06%	1.31%
0.3	avr	93.94%	93.09%
	std	1.41%	2.05%
0.5	avr	94.08%	93.44%
	std	1.49%	0.68%
0.7	avr	94.06%	93.21%
	std	1.06%	0.93%

Table 15. Performance of different amount of levels of dropout in the Recurrent sub-network. Based on 5-fold cross-validation. Used metrics event accuracy, end of season (EOC) accuracy, year 2018

In Table 15 the performance of different levels of the Recurrent sub-network dropout is shown. On average the EOS accuracy remained the same, but the lowest variance was reached with the level of dropout $= 0.5$.

4.4.5 Experiments with the final prediction layers

Size of the middle layer in a dual-layer perceptron was tested. Variants with 150, 300, 500, 1000 neurons were examined that can be seen in Figure 16. The best performance

in terms of event accuracy and end of season accuracy was reached with the size of 500.

Middle output layer size		Event acc	EOS acc
150	avr	87.06%	89.90%
	std	2.05%	0.98%
300	avr	89.16%	90.94%
	std	2.50%	1.25%
500	avr	94.08%	93.44%
	std	1.49%	0.68%
1000	avr	87.99%	90.25%
	std	2.65%	0.87%

Table 16. Performance based on the size of the first layer in a two layer perceptron output. Based on 5-fold cross-validation. Used metrics event accuracy, end of season (EOS) accuracy, year 2018

4.4.6 Activation functions

In the convolutional layers two other activation functions were tested: leaky ReLU and ELU. Their main difference from the rectified linear unit (ReLU) lays in the output of those functions that they give for negative values. When ReLU has an exact zero output for each negative value, leaky ReLU has a linear slope, and ELU has an exponentially shrinking negative slope. For the used leaky ReLU $\alpha = 0.3$ was used, and for ELU $\alpha = 1.0$.

Activation function for convolutional layers		Event acc	EOS acc
ReLU	avr	94.08%	93.44%
	std	1.49%	0.68%
Leaky ReLU	avr	90.11%	91.30%
	std	2.4%	1.33%
ELU	avr	91.41%	91.88%
	std	2.27%	0.60%

Table 17. Performance of different activation functions in a convolutional layers. Based on 5-fold cross-validation. Used metrics event accuracy, end of season (EOS) accuracy, year 2018

Leaky ReLU and ELU allow small gradients when the layer output reaches negative values. According to Table 17 ELU and leaky ReLU activation functions in the convolutional layers experienced a worse performance in both metrics: event accuracy and end of season accuracy. Previously used activation ReLU remained in the model.

4.4.7 Optimizer

Different optimizers were applied to the DeepSense model. All the parameters used are the default for the Keras Tensorflow library. Because the input was unbalanced, the BCE loss was weighted for positive and negative classes. Not all optimizers are appropriate in that case. Those which step is dependent on gradient magnitude may fail (like SGD). RMSprop, Adam, Nadam, Adagrad were tested.

Optimizer		Event acc	EOS acc
RMSprop	avr	87.88%	91.53%
	std	6.67%	1.97%
Adam	avr	85.20%	89.67%
	std	1.37%	0.74%
Nadam	avr	94.08%	93.44%
	std	1.49%	0.68%
Adagrad	avr	78.32%	88.68%
	std	2.22%	0.54%

Table 18. Different optimizer performance. Based on 5-fold cross-validation. Used metrics event accuracy, end of season (EOS) accuracy, year 2018

In Table 18 the tested optimizers are compared. The Adam optimizer was giving low variance in both EOS and event accuracy. At the same time both stable and acceptable results were gained with the help of Nadam optimizer. Nadam is similar to Adam adaptive mechanism but uses Nesterov momentum. RMSprop and Adagrad are also adaptive optimizers, however, those did not outperform the Nadam mechanism.

4.5 Tuned DeepSense model: training on 2017 and testing on 2018

The previous not tuned modification of the DeepSense was struggling to gain a good accuracy when it was trained on 2017 and tested on 2018. We ran this experiment for a tuned model.

	Event acc	EOS acc
tuned	91.09%	89.09%
not tuned	85.94%	88.86%

Table 19. Event accuracy and end of season (EOS) accuracy. Trained on 2017, tested on 2018. The **DeepSense architecture** before and after tuning

In Table 19 it can be seen that the tuned model improved in terms of the event accuracy from almost 86% to 91%. It is known that the fraction of parcels with positive

events in them for the year 2018 approximately equals 88%. It means that the tuned DeepSense model trained on 2017 data was failing to mark correctly those fields that did not have any positive events.

4.6 Summary

The DeepSense model architecture and the purely convolutional network were trained and tested both on 2017 and 2018 data. The testing was based on the 5-fold cross-validation. For the year 2018 the DeepSense got a similar performance as the purely CNN architecture of 90% in term of the end of season accuracy. At the same time, the variance for the end of season accuracy was lower for the DeepSense model. For the year 2017 the DeepSense model gained a similar to the purely convolutional network average event accuracy of 92%. At the same time the variance of the event accuracy for the DeepSense was much lower than for the CNN model. On the merged dataset from 2017 and 2018 the DeepSense model gained similar performance for the end of season accuracy of 90%. The event accuracy appeared to be better for the DeepSense than for the purely convolutional model (90% against 85%).

Afterward, the year 2018 was chosen to tune the DeepSense model architecture. Dataset of 2018 has a substantially bigger number of parcels. It helped to gain more satisfactory results because the DeepSense model has a relatively high amount of trainable parameters. For the experiments the data from 2018 and 2017 was not merged to save the consistency of the weather conditions.

Different model architecture changes were applied for the DeepSense model. It was found that the model gains better behavior without attention mechanism. Instead a simple multi-layer perceptron made of two fully connected layers was used. The model did not improve after adding extra convolutional layers to the Individual and Merged sub-networks. Configuration with one layer per sub-net remained. Each convolutional layer contributes a lot to the model's complexity. Because the amount of data was reasonably limited, the simpler the architecture the lower a chance of the model's over-fitting.

Different sizes of kernels in the convolutional layers were tested. The best performance was found for the kernel [2, 10] in the Individual sub-network and [2, 30] in the Merged sub-network. For the Individual sub-network the kernel with the size [2, 10] was the biggest possible, because the size of an observation window was equal to 10 days. Such a kernel was getting the biggest amount of information covering the longest sequence of days.

Also the number of filters in convolutional layers was explored. The best results remained with 20 filters at each layer. Slightly worse results were obtained with the number of filters equal to 10. But the higher values of 60 filters gave a substantially worse result in both event accuracy and end of season accuracy metrics.

The Recurrent sub-network also was discovered. Different sizes of GRU components were tested. Configuration with 100 units for the first layer and 100 units for the second

outperformed other combinations. It appeared that a smaller amount of units was not enough to let the model generalize well. At the same time too big amounts of 200 units for each layer were also returning worse results, probably due to the model's over-fitting. Also the dropout level between recurrent layers was tested. That parameter was used for regulation purposes. The model performed in terms of the average end of season accuracy similar. But with lower dropout 0.1 and 0.3 the variance was bigger, meaning that it was less stable. Thus, the option with dropout $= 0.5$ was chosen.

The middle layer in the output MLP was tested. The output MLP was making a final prediction of mowing events. The best performance was reached with a size of 500. Other variants were returning significantly worse results in terms of the event accuracy. Probably smaller sizes of 150 and 300 had too small capacity, and with the size of 1000 the model was over-fitting.

Different activation functions in the convolutional layers were explored. In particular, different types of linear units were used: rectified linear unit (ReLU), leaky ReLU, exponential linear unit (ELU). It appeared that small negative gradients that are given by leaky ReLU and ELU functions did not improve the training process. Thus the best performance was reached with the ReLU activation function in the convolutional layers.

Model behavior with different optimizers was evaluated. Nadam optimizer, that is actually the Adam with Nesterov momentum, got the best results for the event and EOS accuracy. Because the binary cross-entropy loss used by the model was weighted, only adaptive optimizers were tested. The Nesterov momentum gave the best performance for the model.

Finally, the tuned model gained 94% event accuracy and 93% end of season accuracy on the 5-fold cross-validation of the 2018 dataset.

After the tuning experiments the tuned DeepSense model was trained on 2017 and tested on 2018 data. Tuning allowed to improve the event accuracy from 86% to 91%. The end of season accuracy remained the same at the level of 89%.

5 Conclusion

In this chapter results are discussed. The conclusion is drowned for the performance of methods that were developed in this thesis work. The discussion about qualities of the models and limitations is held. Future work and perspectives are described.

5.1 Conclusion

The main goal of this thesis was to implement a neural network with DeepSense architecture applicable for farming event detection based on Sentinel-1 and -2 imagery series.

The developed model was compared to a flagship purely convolutional architecture used by Kappazeta. According to the averaged values obtained from 5-fold cross-validation the DeepSense approach showed a better performance for the end of season accuracy metric. The purely convolutional model remained dominant in terms of event accuracy.

The DeepSense model struggled to gain a good accuracy when it was trained on 2017 and tested on 2018 data. Probably it happened due to the small amount of parcels in the training set. During a farming season in 2018 the weather was much warmer and dryer comparing to 2017. Cross-validation on the merged (2017 and 2018) dataset showed that the DeepSense model has a better performance in event accuracy of 90% comparing to 85% gained by the CNN model. It means that the DeepSense model has a better ability to recognize mowing event patterns under the different weather conditions, but it requires a bigger dataset than the pure CNN model.

A set of experiments was conducted to bring the best hyper-parameters of the DeepSense architecture for the farming activity task. It was shown that for the provided dataset of the year 2018 the best configuration of the DeepSense model was as follows:

- one convolutional layer for Individual sub-network,
- one convolutional layer for the Merged sub-network,
- 2 stacked GRU layers of size 100 and 100 for Recurrent sub-network,
- Two fully connected layers as an output for a final event prediction.

Comparing the DeepSense architecture and the purely CNN model advantages and obstacles were found.

The purely convolutional model is more lightweight: it gets trained with a smaller amount of epochs and requires less computational time per each epoch. At the same time the CNN model struggles to omit positive predictions for those fields that did not have any mowing event during a season.

The DeepSense model requires more data to prevent over-fitting. It has a bigger number of trainable parameters and can learn more complex patterns than the purely convolutional architecture. That allows the DeepSense model to perform better in terms of the end of season accuracy. It recognizes more rare patterns of the fields without events and successfully marks them. Overall the DeepSense can be a good general predictor of the season, while the purely CNN model can be used to predict events for each day.

5.2 Future work

The performance of the proposed DeepSense architecture can be improved with a bigger amount of data in the training set. Also the model can be trained on previous years to predict events for a new incoming data. The DeepSense model architecture can be revised with a larger amount of data for training. Moreover, increasing the number of layers for an individual sub-network can help in enhancing the performance.

Another further work direction, can be to explore the development of a combination of the strength of two models and fuse them into a one new model. As the DeepSense architecture model gained better end of season accuracy (93% against 90%) and purely convolutional model reached the best score of event accuracy (96% against 94%), their fused model can be used as a more reliable system for both metrics.

References

[1] Yao, Shuochao & Hu, Shaohan & Zhao, Yiran & Zhang, Aston & Abdelzaher, Tarek. (2017). DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing. 351-360. 10.1145/3038912.3052577.

[2] Rouse, J. & Haas, R. & Schell, J. & Deering, D.. (1974). Monitoring Vegetation Systems in the Great Plains with ERTS. NASA Special Publication. 1.

[3] Liakos, Konstantinos & Busato, Patrizia & Moshou, Dimitrios & Pearson, Simon & Bochtis, Dionysis. (2018). Machine Learning in Agriculture: A Review. Sensors. 18. 2674. 10.3390/s18082674.

[4] Mitchell, Anthea & Rosenqvist, Ake & Mora, Brice. (2017). Current remote sensing approaches to monitoring forest degradation in support of countries measurement, reporting and verification (MRV) systems for REDD+. Carbon Balance and Management. 12. 9. 10.1186/s13021-017-0078-9.

[5] Ryan, Casey & Hill, T.C. & Woollen, Emily & Ghee, Claire Mitchard, Edward & Cassells, Gemma & Grace, John & Woodhouse, Iain & Williams, Mathew. (2012). Quantifying small-scale deforestation and forest degradation in African woodlands using radar imagery. Global Change Biology. 18. 243-257. 10.1111/j.1365-2486.2011.02551.x.

[6] Kussul, Nataliia & Lavreniuk, Mykola & Skakun, Sergii & Shelestov, Andrey. (2017). Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. IEEE Geoscience and Remote Sensing Letters. PP. 1-5. 10.1109/LGRS.2017.2681128.

[7] Zalite, Karlis & Antropov, Oleg & Praks, Jaan & Voormansik, Kaupo & Noorma, Mart. (2015). Monitoring of Agricultural Grasslands With Time Series of X-Band Repeat-Pass Interferometric SAR. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 1-11. 10.1109/JSTARS.2015.2478120.

[8] Tamm, Tanel & Zalite, Karlis & Voormansik, Kaupo & Talgre, Liina. (2016). Relating Sentinel-1 Interferometric Coherence to Mowing Events on Grasslands. Remote Sensing. 8. 801. 10.3390/rs8100802.

[9] Belgiu, Mariana & Csillik, Ovidiu. (2017). Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. Remote Sensing of Environment. 204. 10.1016/j.rse.2017.10.005.

[10] d'Andrimont, Raphael & Lemoine, Guido & Velde, Marijn. (2018). Targeted Grassland Monitoring at Parcel Level Using Sentinels, Street-Level Images and Field Observations. Remote Sensing. 10. 1300. 10.3390/rs10081300.

[11] Stendardi, Laura & Karlsen, Stein & Niedrist, Georg & Gerdol, Renato & Zebisch, Marc & Rossi, Mattia & Notarnicola, C.. (2019). Exploiting Time Series of Sentinel-1 and Sentinel-2 Imagery to Detect Meadow Phenology in Mountain Regions. Remote Sensing. 10.3390/rs11050542.

[12] Kavats, O.O. & Khramov, Dmitry & Sergieieva, Kateryna & Vasyliev, Volodymyr. (2019). Monitoring Harvesting by Time Series of Sentinel-1 SAR Data. Remote Sensing. 11. 2496. 10.3390/rs11212496.

[13] Khabbazan, S., Vermunt, P., Steele-Dunne, S., Ratering Arntz, L., Marinetti, C., & van der Valk, D. et al. (2019). Crop Monitoring Using Sentinel-1 Data: A Case Study from The Netherlands. Remote Sensing, 11(16), 1887. doi: 10.3390/rs11161887

[14] Taravat, Alireza & Wagner, Matthias & Oppelt, Natascha. (2019). Automatic Grassland Cutting Status Detection in the Context of Spatiotemporal Sentinel-1 Imagery Analysis and Artificial Neural Networks. Remote Sensing. 11. 711. 10.3390/rs11060711.

[15] Thang Luong, Hieu Pham, Christopher D. Manning (2015). Effective Approaches to Attention-based Neural Machine Translation. Association for Computational Linguistics. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 10.18653/v1/D15-1166

[16] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. NIPS 2014 Deep Learning and Representation Learning Workshop. arXiv:1412.3555

[17] Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need. arXiv:1706.03762

Appendix

I. Glossary

CAP - common agricultural policy

MSI - multi-spectral instrument

NDVI - normalized difference vegetation index

SAR - synthetic aperture radar

VV - vertical transmit, vertical receive polarization

VH - vertical transmit, horizontal receive polarization

RNN - recursive neural network

GRU - grated recursive unit

CNN - convolutional neural network

MLP - multi-layer perceptron

EOS - end of season

ReLU - rectified linear unit

ELU - exponential linear unit

II. DeepSense model architecture variant

Layer	Specification
Input	2 sensors dimension 2x10x15
Individual sub-network	
Convolutional layer 1	kernel [2, 10] stride [1,1] padding valid filters 20 activation ReLU
Batch normalization	scale =false
Flatten	
Concatenation	2 sensors concatenated by new dimension
Merged sub-network	
Convolutional layer 2	kernel [2, 30] stride [2,1] padding valid filters 20 activation ReLU
Batch normalization	scale =false
Flatten	
Concatenation	each time window concatenated by new dimension
Recurrent sub-network	
GRU layer 1	units 100 activation function tanh recurrent activation sigmoid
Dropout	level of dropout 0.5
GRU layer 2	units 100 activation function tanh recurrent activation sigmoid
Attention layer	implementation from [15] final dense layer of size 500 with tanh activation
Output layer	dense fully connected size 150 activation sigmoid

Table 20. DeepSense model architecture variant with specifications of layers

III. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Kyrylo Medianovskyi**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Exploring DeepSense Neural Network Architecture for Farming Events Detection,

   supervised by Amnir Hadachi.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Kyrylo Medianovskyi
15/05/2020