UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Viacheslav Komisarenko

Farming events detection from Sentinel-1 and -2 satellite imagery time series with deep learning

Master's Thesis (30 ECTS)

Supervisor: Sherif Sakr, PhD
Supervisor: Kaupo Voormansik, PhD
Supervisor: Yousef Essam, MSc

Tartu 2019

Farming events detection from Sentinel-1 and -2 satellite imagery time series with deep learning

Abstract:
Satellite imagery allows building applications in a variety of domains. Agriculture is an example with a lot of possibilities for automation. Thousands of inspectors visit fields across the European Union to check if mowing events were performed. Reliable automated detection system can free this work force for other needs.

Sentinel-1 (coherence in VH and VV polarization) and -2 (normalized difference vegetation index) data was chosen as the base feature set in this thesis. Convolutional neural network model was created which is capable to detect mowing events based on satellite's imagery time series. Using Estonia 2018 labeled data about 2000 fields, the model was trained and optimal configuration of hyperparameters was tuned.

Transfer learning techniques were applied based on Swedish 2018, Danish 2018 and Estonian 2017 data. Weights of trained models were re-used to improve performance on target Estonian 2018 dataset. Based on the reject region method, an algorithm for finding a subset with highly confident and accurate predictions was proposed.

Proposed modifications allowed to obtain event accuracy of 76.1% and end of season accuracy of 96.6%. The proposed model architecture is suitable for practical use in the mowing detection system.

Keywords:
Satellite data, Sentinel-1, Sentinel-2, CNN, convolutional neural networks, farming events detection, transfer learning, reject region.

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

Põllumajandustegevuste tuvastamine Sentinel-1 ja -2 satelliidipiltide aegridadelt süvaõppega

Lühikokkuvõte:
Satelliitmõõtmised võimaldavad arendada rakendusi paljudes valdkondades. Põllumajandus on heaks näiteks rohkete satelliitseire põhiste automatiseerimisvõimalustega. Põllumajandustoetuste jaotamise järelevalvet tehakse tänini peamiselt välitöödega, Euroopa Liidus on sellega hõivatud kümneid tuhandeid inspektoreid. Usaldusväärne automaatne satelliitseire põhine kontrollisüsteem võimaldaks selle töö jõu vabastada ja suunata kõrgema lisandväärtusega sektoritesse.

Sentinel-1 VH- ja VV-polarisatsiooni koherentsus ja Sentinel-2 vegetatsiooniindeks NDVI moodustasid käesoleva töö algse tunnuskomplekti. Töö raames arendati välja sidumnärvivõrgu mudel niitmise tuvastamiseks satelliitmõõtmiste aegridadest. Mudeli sobitamiseks ja parimate hüperparameetrite leidmiseks kasutati enam kui 2000 Eesti rohumaa märgendatud andmeid 2018 suvest.

Siirdeõppe meetodite testimiseks kasutati Rootsi 2018, Taani 2018 ja Eesti 2017 märgendatud andmeid. Eeltreenitud mudeleid taaskasutati Eesti 2018 sihtandmekogu peal täpsuse suurendamiseks. Kõrgema usaldusväärsusega ja täpsemate tulemuste saamiseks pakuti välja praaktsooni põhine algoritm. Töö käigus leitud täiustusega saavutati 76,1% üksiksündmuste tuvastamise täpsus ja 96,6% põllupõhine niitmata põldude tuvastamise kogutäpsus hooaja lõpuks. Välja pakutud mudel sobib praktiliseks kasutamiseks niitmise tuvastamise infosüsteemis.

Võtmesõnad:
Satelliitseire, Sentinel-1, Sentinel-2, CNN, sidumnärvivõrk, põllumajandustegevuste tuvastamine, siirdeõpe, praaktsoon.

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

Contents

1 Introduction

Farming event detection is a task of determining agricultural activity (e.g. cutting, ploughing, cultivation events) or its absence. One of the main reasons behind the need for monitoring activity on fields is the subsidies program introduced by the European Union's (EU) common agricultural policy (CAP)[1]. The main idea of CAP is that farmers are supported for managing the agricultural fields in an environmentally friendly way. It includes, depending on the country, regular mowing, growing only specific crops types etc. Mowing events monitoring is the most relevant to the subsidy program in Estonia.

To check if farmers comply with the agreed conditions, inspectors of the paying agency (PA) carry out field visits. Depending on the country there are hundreds to thousands of inspectors. Using satellite data their work can be partly or fully automated, saving from the work force costs and freeing the people to higher valued jobs.

Sentinel-1 and -2 are Earth Observation (EO) satellites. Sentinel-1 carries a synthetic-aperture radar (SAR) while Sentinel-2 is taking optical images. Satellite imagery is a regular, reliable and fast developing source of data. It is widely used for agricultural, marine, emergency management and other applications. In agriculture, the dense time series of S1 and S2 imagery enable to follow the phenological changes of each crop and the farming practices on every parcel.

Deep learning is a popular tool for classification problems. Besides this, neural networks are actively used for time series analysis tasks. Given large amount of data, deep learning methods outperform traditional statistical and machine learning techniques. Despite the fast progress in developing new techniques, there is still no universal recipe on how to build a model that could solve any task. Applying deep learning to each problem needs still an individual approach.

The main goal of the thesis is to create a robust algorithm for farming events detection (on the example of mowing as most relevant to the subsidy program in Estonia) detection based on Sentinel-1 and -2 imagery time series. Based on the data and pre-processing capabilities provided by OÜ KappaZeta, neural networks model was created, different aspects of architecture, optimization process and feature engineering were studied to reach high accuracy in mowing events detection. OÜ KappaZeta is an Estonian company that is focused on development Earth observation applications and services in the agricultural domain.

Chapter 2 gives an overview of related works: papers that used similar features, supervised learning methods and problems from the same domain.

Chapter 3 provides a detailed description of gathering, pre-processing and labeling the data. It includes subsections about Sentinel-1, Sentinel-2 and derived features as well as field books and manual labeling. Also, an outlier detection algorithm is proposed in

[1]The common agricultural policy at a glance: https://ec.europa.eu/info/food-farming-fisheries/key-policies/common-agricultural-policy/cap-glance_en

this section.

Chapter 4 covers all information regarding the model setup and hyperparameters tuning. To construct the model, a number of layers, their types, internal parameters, optimizer and learning rate were chosen in order to maximize performance on the target dataset.

Chapter 5 includes experiments with transfer learning methods. The main aim was to re-use knowledge obtained from training models in different countries and seasons. It also introduces the reject region - a technique that allows selecting fields with only highly confident predictions.

2 Related works

The topic of events detection using satellite's data has become widely known in the research community since the launch of the Copernicus program [2]. Besides agriculture, satellite imagery is used in emergency mapping [1], urban areas [2], marine monitoring [3] etc.

There is a limited amount of unique satellite-based features which are sensitive to vegetation and agricultural events. One of the main examples of such feature is the normalized difference vegetation index (NDVI), which could be obtained from Sentinel-2 imagery. It is an established indicator (firstly introduced in 1974) for monitoring vegetation systems [4]. The use of SAR repeat pass coherence in VV (vertical transmit, vertical receive) and VH (vertical transmit, horizontal receive) polarization is relatively new for agricultural applications. The statistical significance of the median coherence increase after mowing events was shown in [5]. In this work, the authors analyzed coherence obtained from 12-day repeat cycle Sentinel-1 image pairs. The same coherence calculation, but with 6-day repeat cycle combining Sentinel-1A and -1B imagery was used in this thesis. Convolutional neural networks (CNN) are actively used for classification tasks. One of the first works in this topic was performed already in 1995 [6]. However, time series were not the main focus of the work, and the number of layers was too small to consider the network as deep. Deep convolutional neural network on multivariate time series was studied in [7].

Crop classification is another topic of active research in the domain of agricultural applications using satellite data and machine learning techniques. Sentinel-1 and Landsat-8 data were used to create multi-layer perceptrons model for crops in Ukraine [8]. Further work of the authors explores the potential of convolutional neural networks for land cover and crop classification based on nineteen multitemporal scenes acquired by Landsat-8 and Sentinel-1A RS satellites [9]. Authors achieved 85% accuracy for prediction of all major crop types. Sentinel-2 was used for urban change monitoring in [10]. Authors proposed an architecture based on CNN and achieved 90% accuracy of urban change detection using Onera Satellite Change Detection dataset. Combination of Sentinel-2A and ALOS-2 PALSAR-2 imagery was used to estimate forest biomass in Iran [11]. Proposed Multi-Layer Perceptron model achieved R^2 score 0.44.

Ship detection using Sentinel-1 data was studied in [12]. Proposed complex CNN-based architecture had F1 score 0.87. Sentinel-1 and -2 imagery was used for land cover classification model in Sweden and up to 98% accuracy was achieved [13]. Besides crop classification, crop yield forecasting based on satellite data is another field of research with high interest. Bayesian neural networks were applied on top of normalized difference vegetation and enhanced vegetation indices to predict crop yield amount in [14].

[2] About Copernicus: https://www.copernicus.eu/en/about-copernicus

3 Dataset

This chapter gives an overview about data used with detailed description about the meaning, limitations and capabilities of the features. Processing methods like smoothing and outlier detection are also discussed. An explanation of the target variables and their derivation is given.

The data segment chosen for analysis is Estonia 2018, which has around 2000 fields in total. In transfer learning subsection Swedish and Danish 2018 data was used (transferring across countries) and Estonian data of 2017 (transferring across years). The size of Swedish and Danish set is 4000 fields, Estonian 2017 set has about 1500 fields.

3.1 Features

Base feature set was obtained from Sentinel-1 and -2 satellite missions. It is a part of Copernicus program, which provides freely open Earth observation data. However, specific tasks need dedicated data pre-processing techniques. The feature set derivation from raw satellite data is not a subject of this thesis and will not be described here. Data processing was performed by Kappazeta OÜ.

3.1.1 Sentinel-1 features

Sentinel-1 satellite carries a synthetic-aperture radar (SAR) instrument. SAR data has high resolution and its acquisition is independent from the weather conditions. Coherence (in both VV and VH polarization) was chosen from SAR features because it is shown [5] to be sensitive to changes in vegetation and agricultural events. Coherence is a normalized measure of similarity between two consecutive (same relative orbit) SAR images. In case mowing event happens between two SAR images, they will be dissimilar which will result in low coherence. But right after mowing the coherence will remain slightly higher for a period of time, because there is less vegetation causing temporal decorrelation. Tall unmown grass on the other hand will result in low coherence, because of temporal decorrelation caused by the changes in the vegetation layer.

Coherence in VH polarization (cohvh), coherence in VV polarization (cohvv) as numeric features have the following characteristics:

- Values are within 0 and 1 range;
- Measurements are regular and periodic (because of regular acquisitions independent from cloud cover and sunlight);
- Values could sometimes be noisy - it comes from the weather effects (wind and rain as well as high and variable moisture content in some of the soils - e.g. grasslands on marshland and river valleys).
- The signature of mowing event - temporary sharp increase of coherence.

3.1.2 Sentinel-2 features

Sentinel-2 provides optical imagery. The main difference with Sentinel-1 is weather dependency. It means that clouds will be visible on the image. Sentinel-2 data is provided by the European Space Agency (ESA) together with a cloud mask, which is able to filter clouds on the image with moderately good accuracy. The chosen Sentinel-2 feature is NDVI (normalized difference vegetation index) - indicator that is related to the amount of live green vegetation. It is a well established and probably the most widely used optical remote sensing feature for agricultural applications. The signature of mowing event is rapid decrease of ndvi.

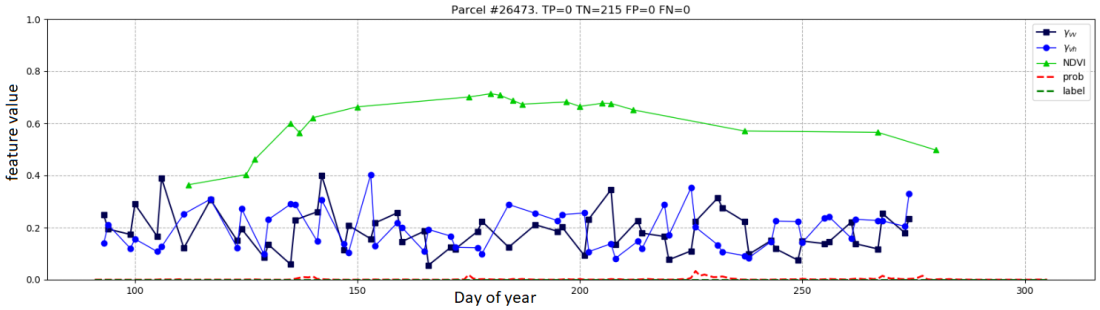

Figure 1. Not mown field during whole season.

A typical signature of not mown field is in Fig. 1. The x-axis represents day of year, while y-axis has no units and a scale from 0 to 1. Ndvi measurements are green, cohvh and cohvv are blue and black respectively (denoted as γ_{vh} and γ_{vv} - accepted notation in remote sensing community). For not mown field, the typical signature of ndvi during the season is a half-oval curve; coherence is shaky but remains at approximately the same level, without clear trend changes.

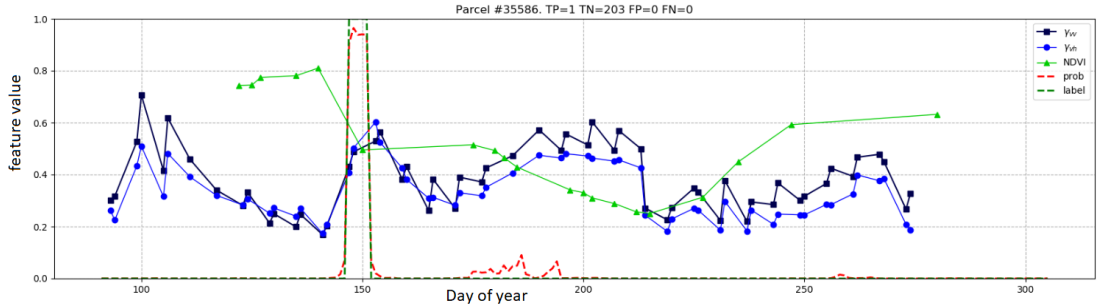

Figure 2. Field with one mowing event

Fig. 2 contains an example of a field with one mowing event during the season. The legend for the plot is the same as for Fig. 1. Around day 150 we have a spike marked with red and dark green colors. It is a mowing event: both cohvh and cohvv increase rapidly while ndvi goes down sharply. A similar signature is around 40 days later, however, most probably it is not a mowing event - increase for coherence and drop for ndvi are not as fast, possible explanation is a summer draught.

Ndvi as a numeric feature has the following characteristics:

- Ndvi has a value in a range from 0 to 1.
- The same as all Sentinel-1 features, ndvi value for specific field is a mean aggregation of pixel values within this field; event performed at only part of the field will be hard to detect based on current feature set.
- Sentinel-2 images are obtained regularly, at Estonian latitude there are two new images in every 5 days about each geographical location. However, valid ndvi measurements are irregular and quite sparse. Due to cloud cover, around 75% of total measurements are invalid in Estonia, and slightly lower fraction in Southern Sweden and Denmark;
- The official ESA cloud mask is not ideal: it filters out most of the cloud covered areas, but not all. NDVI, which is computed based on cloudy imagery results in outliers of the time series. The signature of the outlier is a suspiciously low ndvi value in particular day while previous and next values are quite high.

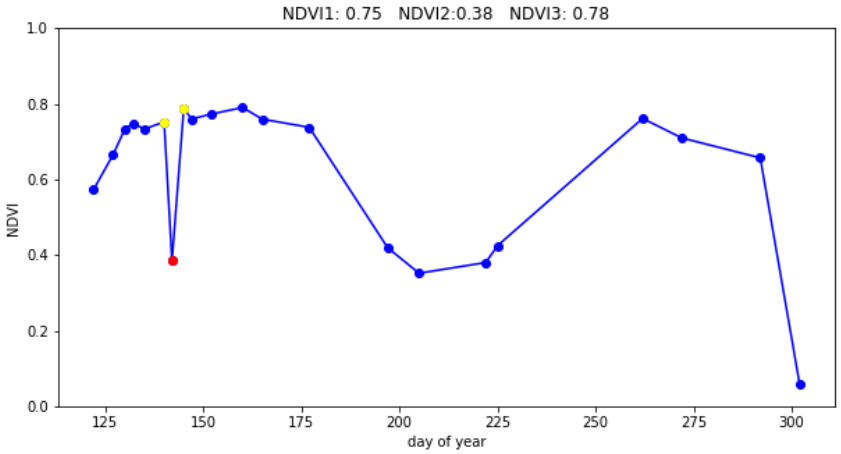

Figure 3. Outlier in ndvi measurement.

Fig 3. contains example of an outlier in ndvi measurement. Y axis is 0-1 scale for ndvi, x axis is day of the year. An outlier is marked with red dot, nearest previous and

following measurements are marked with yellow color. On top of the plot, these three exact values are given and it could be seen how large is the gap for an outlier.

It is similar with mowing event signature, but ndvi after fast drop is recovering during 10-20 days for mowing event [15] while recovering in outlier's case is immediate. Additional outlier detection techniques on top of the cloud mask could be introduced to catch suspiciously low values.

NDVI outliers detection method

To develop a reliable ndvi outlier's detector, it is needed to understand the data. Outliers could be detected by checking previous and next values. We created and tested a method that iterates through all possible ndvi triplets (three consecutive ndvi measurements for a given field) and checks the difference between second and first values as well as between third and second. The signature of an outlier is highly negative first difference and highly positive second. It is also important to note time difference among measurements, as they could be quite sparse: signatures with time difference lower than 10 days are outliers with high probability, while for larger time difference actual mowing events could have the same signatures.

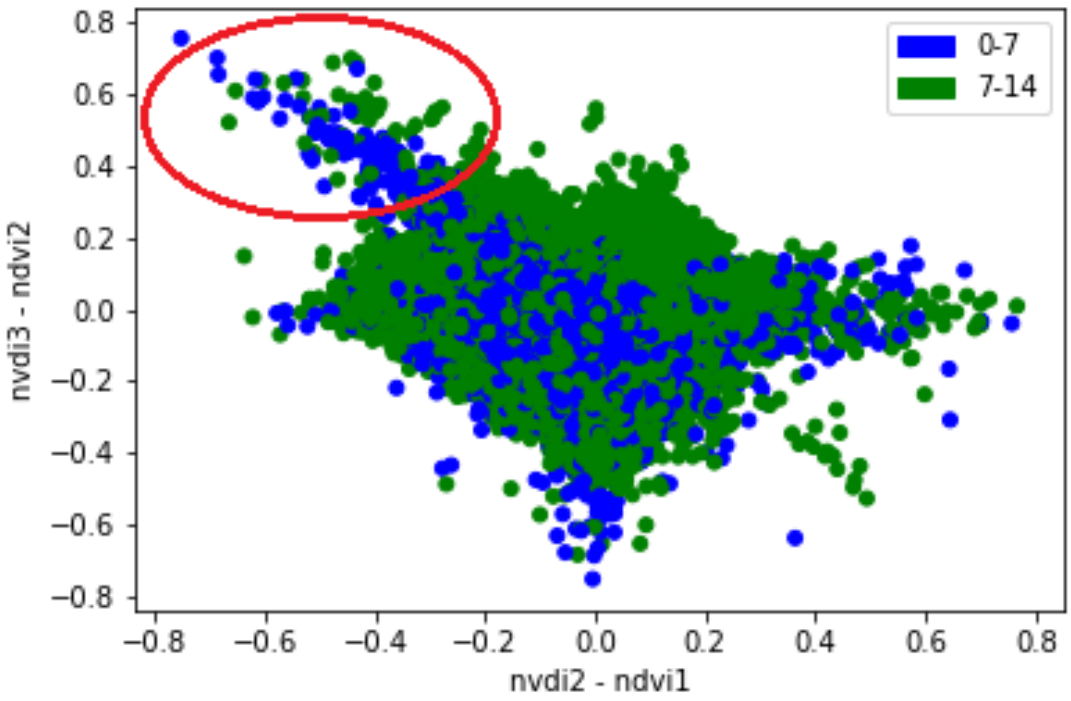

Figure 4. NDVI triplets scatter plot.

Fig. 4. contains a scatter plot of all ndvi triplets in the data. Y axis is a difference

between third and second values in the triplet while X axis is a difference between second and first. Time frame in days among three measurements are written in the legend: triplets with up to 7 days difference are blue, with difference from 7 to 14 days are green. The structure of points is interesting - it looks like a rhombus with small cloud of possible outliers in upper left corner (denoted with a red oval in the figure). The signature of outliers and this cloud matches, to filter from the list actual mowing events we should only consider triplets within up to 10 days interval (as the mowing event signature is recovering for at least 10 days). Knowing rhombus equation (the center is approximately in (0, 0), axis are parallel with correspondent plot axis and the side length is around 0.6), the filtering rule could be easily constructed:

$$ndvi\_3 - 2 \cdot ndvi\_2 + ndvi\_1 \geq 0.6 \tag{1}$$

In Formula 1, ndvi_1, ndvi_2, ndvi_3 are consecutive measurements within 10 days interval. All triplets that match this equation contains an outlier on second position - the ndvi_2 should be removed. Around 0.1% of ndvi measurements were deleted from the data when applying this outlier filtering rule.

3.1.3 Additional features

Based on existing Sentinel-1 and -2 features we could derive new features that could improve training process.

One example of an additional feature is a meaningful combination of existing ones. Cohvh, cohvv have often similar behaviour but sometimes different as well. New feature could be introduced that will try to capture overall coherence trend. Preferably it should be a non-linear combination of cohvh and cohvv, because the neural network is capable of discovering effective linear dependencies of features inside respective layers. One of the simplest options, quadratic mean of cohvh and cohvv was added to features list as mixed_coh. The formula for mixed_coh is the following (Formula 2):

$$mixed\_coh = \sqrt{cohvh \cdot cohvv} \tag{2}$$

Other cross feature dependencies turned out to be not as meaningful.

Smoothing is a possible processing step for noisy features. We could use either only smoothed version of features or keep initial values also. In current task, cohvh and cohvv are the features that could be smoothed. The following techniques were tried in order to choose the best fit for the task: exponential moving average (EMA), Kalman filter, moving average. Tests showed that the performance is similar with every mentioned method but setup with exponential moving average has slightly higher event and end of season accuracy. EMA has one parameter α that indicates how much weights should be given to past measurements. An exact recursive expression for EMA applied to cohvh (the same equation for cohvv) is described with the next Formula 3:

$$EMA(cohvh_n, \alpha) = \alpha \cdot (cohvh_n) + (1 - \alpha) \cdot EMA(cohvh_{n-1}, \alpha) \tag{3}$$

It is assumed in the equation that $EMA(cohvh_0, \alpha) = cohvh_0$. Experiments showed that parameter $\alpha = \frac{1}{3}$ has the best performance in terms of metrics mentioned above.

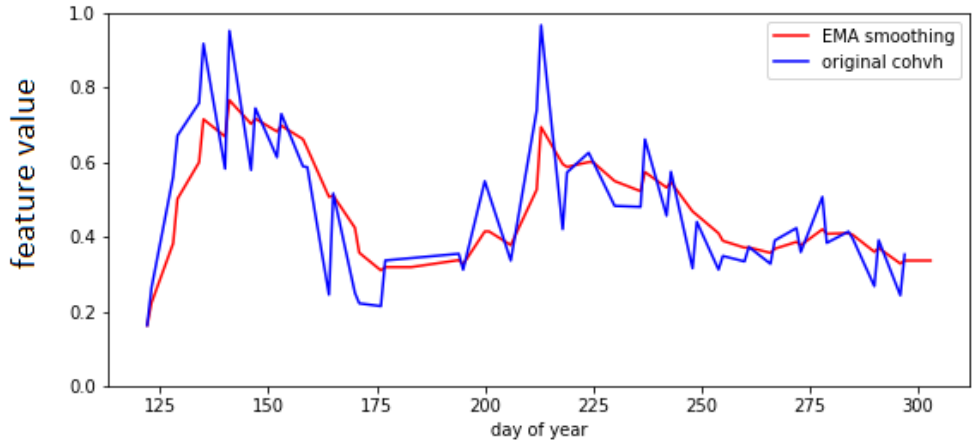

Figure 5. Example of EMA smoothed cohvh.

An example of EMA smoothing is in Fig. 5. The EMA smoothed cohvh_sm and cohvv_sm were added to list of features.

Each obtained measurement has its date. Date could be an important feature for the model to adapt, as it is more likely to have mowing events in the summer rather than in early spring (at least for Estonia). Date, or normalized day of the year, was included into total features list (named as 't'):

$$t = \frac{day\_of\_year}{365} \tag{4}$$

The value of 't' could be in [0; 1] range. We also introduced another time feature that pays attention to gaps in time series. It was called 'dt' and equals to normalized difference in days from current measurement to previous one. Normalization was performed with min-max scaling:

$$dt = \frac{diff - min\_diff}{max\_diff - min\_diff}, \tag{5}$$

where min_diff and max_diff in Formula 5 are minimum and maximum difference in days from previous measurement obtained from training data.

It is known that the signature of the mowing event is an increase of coherence features and sharp decrease of ndvi. It is therefore logical to introduce the difference features and/or slopes of the feature curves. Both differences and derivatives were studied. It could be noticed that if we approximate slope with backward first-order difference

equation, then it could be expressed via division of the difference and dt. To sum up, the following three derived features were added to the list: ndvi_diff, cohvv_sm_diff, cohvh_sm_diff, ndvi_derivative, cohvh_sm_der, cohvv_sm_der.

Each Sentinel-1, Sentinel-2 measurement is obtained from a specific orbit. Orbit could be determined via its RON (relative orbit number). RON defines the area, the time moments and specific incidence angles where data will be acquired. This information could be included to the model but it has no direct connection with agricultural events detection. Initial experiments showed that the model performs worse with this feature and it was not added.

The total list of features is in Table 1.

Table 1. Total list of features.

№	Name	Feature description
1	ndvi	Normalized difference vegetation index, obtained from Sentinel-2.
2	cohvv	Coherence in VV polarization, Sentinel-1 feature.
3	cohvh	Coherence in VH polarization, Sentinel-1 feature.
4	t	Normalized day of year when measurement was obtained.
5	dt	Normalized difference in days from previous measurement.
6	cohvv_sm	Smoothed cohvv with EMA (with parameter $\frac{1}{3}$).
7	cohvh_sm	Smoothed cohvh with EMA (with parameter $\frac{1}{3}$).
8	mixed_coh	Quadratic mean of cohvv and cohvh.
9	ndvi_diff	Difference from previous ndvi measurement.
10	cohvv_sm_diff	Difference from previous cohvv_sm measurement.
11	cohvh_sm_diff	Difference from previous cohvh_sm measurement.
12	ndvi_derivative	The slope of the line between previous and current ndvi value.
13	cohvh_sm_der	The slope of the line between previous and current cohvh_sm value.
14	cohvv_sm_der	The slope of the line between previous and current cohvv_sm value.

Dataset is constructed by merging Sentinel-1 and Sentinel-2 features (together with additional features). If specific date has only one type of measurements (either Sentinel-1 or -2), missing values from the other type will be linearly interpolated with nearest valid measurements of this type.

3.2 Labels

There were two main sources for labelling: field books and manual labelling. Each source has own advantages and disadvantages.

3.2.1 Field books

Farmers record dates and descriptions of farming activities into field books. There are a lot of electronic alternatives for filling necessary data, but paper-based books are still preferred option among many farmers.

OÜ Kappazeta has agreements with farmers about providing field books for gathering data and further analysis. Starting year of agreement was 2017. In 2017, Kappazeta gathered data about more than 1500 fields and inserted manually into the database. From 2018 there were information about more than 2000 fields.

Other source of labels was field visit reports of Estonian paying agencies (PA) inspectors. However, total number of visited parcels was rather small in comparison with other sources, and mainly inspectors approve or not information that was already stored in field books. Most useful information from the inspectors is when the field was not mown at all during whole season. Such fields could be sanctioned for non-compliances; their accurate detection is one of the main goals of this work.

Manual comparison of field book data with feature set time series brought some useful observations:

- There are up to 7 days shifts between recorded farmers' data and visible signatures; It could be that farmers fill their field books in the end of the season by memorizing approximate dates of events instead of providing accurate information;
- Sometimes human experts have not found any visible signatures in provided dates. There could be different explanations: farmers made mistakes in dates; event was performed only on part of the field and it changed aggregated signature slightly; field is very small and computed features have a large variance.

3.2.2 Manual labelling

In general, it is possible to detect clear mowing events visually. If there is a sharp coherence increase with a ndvi decrease and there are enough data points to support it, an expert could mark this signature as a mowing event. The exact date of the event was agreed to be marked one day before the beginning of the signature.

In total, 1000 fields were manually labelled in each of the following areas (Sweden North, Denmark North, Sweden South, Denmark South, Estonia) given time series from 2018.

There are advantages and disadvantages of manual labelling in comparison with field books:

- Manual labelling will always have clear signature of an event while field books are quite noisy; the perfect model cannot be more accurate than human expert. If experts will not mark weak signature events, model will not pick them up also.

- The model could be biased compared to real data; however, there are less erroneous events in the training set.

3.3 Summary of the chapter

The basic features obtained from Sentinel-1 and -2 imagery time series are ndvi, cohvv and cohvh. The typical signature of a mowing event is the temporary sharp increase of cohvh, cohvv and the rapid drop of ndvi. Additional features were derived that includes features based on the date, smoothing, differences and derivatives of the basic features. In total, 14 features will be used for analysis. Also, an outlier detection method for ndvi was proposed.

There are two ways target variables could be obtained: field books and manual labeling. Estonian 2017 and 2018 datasets are based on field books, Swedish and Danish 2018 were manually labeled. Only Estonia 2018 data will be used in the chapter about model construction, while others will be used in the transfer learning chapter.

4 Model construction

This chapter provides a detailed description of all model construction steps. To justify choice on each step, a set of experiments were performed. The results of the experiments are included in this section.

4.1 General setup

One training sample is a data about a specific field during one season. It means that the sample is a matrix where the first dimension is a time and the second dimension is a specific feature. By combining matrices of all fields, we receive a three-dimensional tensor - the whole dataset.

The target variable of the sample is a vector, which consists of binary statuses (mowing or not) at each point of time. The time dimension of one sample is limited to a year (one season). The architecture which will map input to output will be based on convolutional neural networks (CNN), more specifically, one-dimensional convolutional neural networks (1-D CNN).

Convolutional layers are the main building blocks of CNN. The output of the layer is the convolution of input with trainable kernels. Main hyper-parameters are the number of filters (number of different kernels) and kernel size.

The 1-D term means that the filter, or feature detector, slides the data only in one dimension - time. Filters are moving in both directions of time, which allows model to use future data to predict past mowing events. On the one hand, it ensures better quality of predictions. On the other hand, this approach is not applicable for real-time mowing event detection. However, it is possible to predict events with a lag of several days. Exact size of the lag should be greater than a half of the CNN window length.

The length of the CNN sliding window is constant in terms of number of measurements. It is logical to have the constant length also in the time dimension, which helps not to confuse the model. To receive constant window length in time, it is enough to have the same interval between measurements. With our feature set it is not the case by default because of irregular Sentinel-2 features. To guarantee an equal interval, 1-day grid was introduced and all missing values were linearly interpolated. In practise, it means that plots in Fig. 1-2 remains the same but values are extracted from each date.

The general idea of the linear interpolation for the 1-day grid is demonstrated in Fig. 6.

Given that we are interested only in observations during the season, it is not needed to record winter measurements. More specifically, as the vegetative season is approximately the same across all Estonia, the model will operate only with data from April till October. The input size will be fixed by the number of features and the length of the season - 215 days will be considered (some of the days will consist of real measurements while others

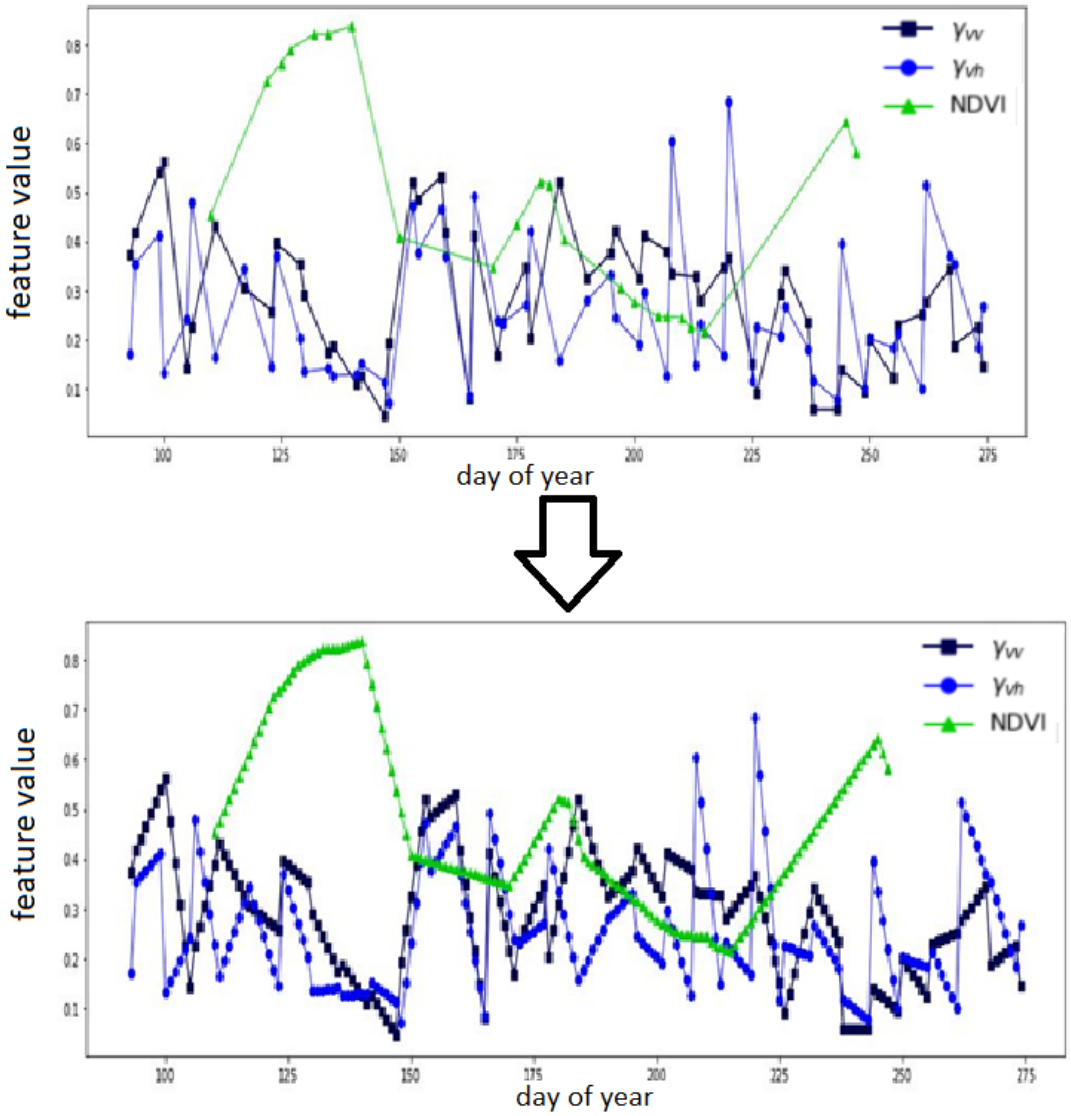

Figure 6. Feature set time series before and after linear interpolation.

are linearly interpolated). The output of the model is a binary status for each out of 215 days.

The training process will be organized with random mini-batches generation. Each batch will consist of a data from 5 random fields, and the number of batches during the epoch will equal to the number of fields. In average, every field will be considered 5 times during an epoch. The general scheme of architecture is given in Fig. 7. The model consists of three 1-D CNN layers. Between first and second, as well as between second

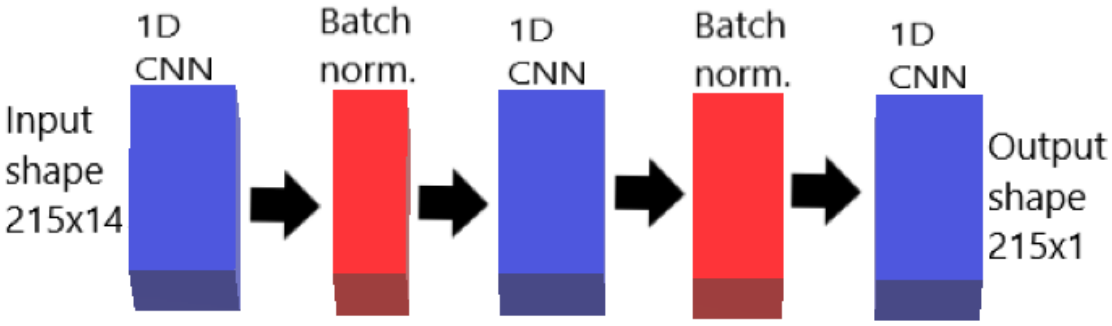

Figure 7. Baseline model architecture.

and third layer there is batch normalization with default parameters. It is a technique that increases stability of neural network and has regularization effect. Batch normalization layer scales and normalizes an output of previous activation layer. Detailed description of each 1-D CNN layer is listed below:

1. 35 filters, 20 kernel size, same padding, sigmoid activation, Xavier weight initialization, clip value 10.

   Activation function is the function applied to the output of the layer to introduce non-linearity to the network. Sigmoid activation is given by the following Formula 6:

   $$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

   Sigmoid transformation is applied point wise.

   Initial weight values are assigned randomly following some distribution. Proper choice of the distribution allows network to train faster, as the initial weight values will be closer to the optimum. Besides this, Xavier initialization [16] is capable to deal with exploding and vanishing gradients problem.

   Gradient clipping is another way to solve exploding gradients problem. Clip value 10 means that gradient coordinates greater than 10 (lower than -10) will be clipped to 10 (-10).

2. 25 filters, 10 kernel size, same padding, sigmoid activation, Xavier weight initialization, clip value 10.

3. 1 filter, 10 kernel size, same padding, sigmoid activation, Xavier weight initialization, clip value 10.

The optimizer is Nadam with default parameters (beta_1=0.9, beta_2=0.999, epsilon=None, schedule_decay=0.004), except learning rate, which equals to 0.001.

The loss function is a binary cross entropy, which is widely used for classification tasks, where the target variable could only have two different values.

Monitored metrics will be described in the subsection Evaluation.

4.1.1 Labelling

The target variable (mowing) can take two values: zero (not mown) or one (mown). The label was set as mown at the actual start day and 6 consecutive days after it, otherwise it was 0 (not mown). This configuration showed significantly better performance than several alternatives (e.g. cumulative labelling, when label at each date shows how many mowing events happened before this date, or vector-like labelling, where first coordinate is responsible for the first event, second coordinate for second and up to the maximum number of events).

4.1.2 Evaluation

To evaluate performance of the model, the following metrics were used:

End of season (EOS) accuracy - if the model predicted mowing event (predicted probability was above 50%) at least once during the season, the field is considered as predicted as mown, otherwise - predicted as not mown. These binary outcomes are compared with actual data and end of season accuracy is calculated. End of season recall and precision of the best models will also be studied, but accuracy will remain the main end of season metric, expressed with one number.

Event-based accuracy - this metric has more complex definition but the meaning is expressed by the name - accuracy of event detection.

Event-based accuracy will be defined from true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) of event detection.

TP - number of times mowing events were predicted correctly. To consider predicted and actual events as matched, start day of the predicted mowing event should be not more than 3 days earlier and not more than 6 days later than actual start day of the mowing event. It might still happen that within these 9 days there are several predicted mowing events. To deal with it, only the first is counted as the TP, every next one is considered as a FP. Based on definition, this number could be from 0 to actual number of events for each particular field.

TN - is the only point-wise component here. It is normalized index of how much times model's negative output (not mown in specific date) is equal to actual negative label. Normalization constant is equal to 0.01. TN is not present in precision and recall equations, but is used for accuracy calculation. However, TN value is approximately the same for all of the models performing at least moderately, changes are really minor in comparison with an impact of TP, FP, FN for accuracy.

FP - number of times mowing events were predicted incorrectly. It includes two scenarios: predicted start of mowing event does not fit into a 9-days frame (explained above) with the actual start of some mowing event; it is second or later predicted event that matches a specific actual event. A number of FP could be from 0 to number of predicted events. FP + TP is equal to total number of predicted events.

FN - number of times actual mowing events were not catched (see definition of TP). FN for the particular field could only be in range from 0 to actual number of mowing events. Also, TP + FN should equal to actual number of events.

Event-based accuracy was computed with Formula 7.

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

It often happened that improvement in event based accuracy led to decrease in end of season accuracy or vice versa. In order to choose the best model, final criteria is the mean of the end of season and event-based accuracy. Each model configuration was run 5 times, after that mean accuracy and standard deviation were computed. One model was assumed to be better than other, if the difference between mean accuracies was greater than two standard deviations, otherwise model's performances are on the same level.

4.1.3 Baseline scores

There are two baseline scores for the current thesis.

The first baseline is the accuracy of long-short term memory (LSTM) neural network architecture on the given dataset, which was operating in OÜ Kappazeta in 2018.

The second baseline is the accuracy of 1-D CNN based model, which was developed in OÜ Kappazeta in 2018. Models were trained on Estonian 2018 dataset. Train, validation and test sets were split randomly with fractions 64%, 16% and 20%. Baseline accuracies are listed in Table 2.

Table 2. Baseline performance.

Baseline	**Event accuracy, %**	**End of season accuracy, %**
LSTM	62.4	79.5
CNN	69.4	96.0

The main goal of the thesis is to create the model that will outperform baseline performance in terms of any of these metrics.

4.2 Features engineering

The full list of initial features (Sentinel-1, Sentinel-2 and additionally derived) was described in Dataset section. In this subsection, an optimal subset of features will be

chosen based on features importance techniques.

There are a lot of features importance techniques, but most of them can not be applied for the current task, as we have matrix-like input and vector-like output. The general idea of the chosen technique is that after random shuffling of an important feature value of loss function will be much worse, while shuffling of an uninformative feature will not influence loss function a lot. Given loss function's increase after shuffling each feature separately we could rank features by their importance. After that we dropped the least important feature and repeated the same procedure again, while only the most important feature was left. Obtained list of features ordered by importance (from the most important to less) together with importance scores is in Table 3.

Table 3. Features importance.

№	Name	Importance score
1	ndvi	0.12
2	mixed_coh	0.08
3	cohvv	0.07
4	t	0.05
5	cohvv_sm	0.04
6	cohvh	0.04
7	cohvh_sm	0.04
8	ndvi_diff	0.02
9	cohvv_sm_diff	0.02
10	cohvh_sm_diff	0.02
11	dt	0.01
12	ndvi_derivative	-3e-10
13	cohvh_sm_derivative	-3e-10
14	cohvv_sm_derivative	-3e-10

The explanation of the features listed above is performed in the end of subsection Features. The list shows that original features are mainly better than derived. Differences and derivatives are the worst. However, mixed_coh has the second score and the normalized time is on the fourth position. The ndvi score was approximately 50% higher than the second score.

To choose the optimal subset of features, we will run the model with all features, after that the worst features will be removed and the model will be trained again, after that the training process will be performed without two worst features and so on, until only one feature (ndvi) will be left. In accordance with Evaluation subsection, metrics are end of season accuracy and event-based accuracy.

The Fig. 8 shows the mean and the standard deviation of EOS accuracy for training with different number of best features. It is noticeable that setup with ndvi and mixed_coh

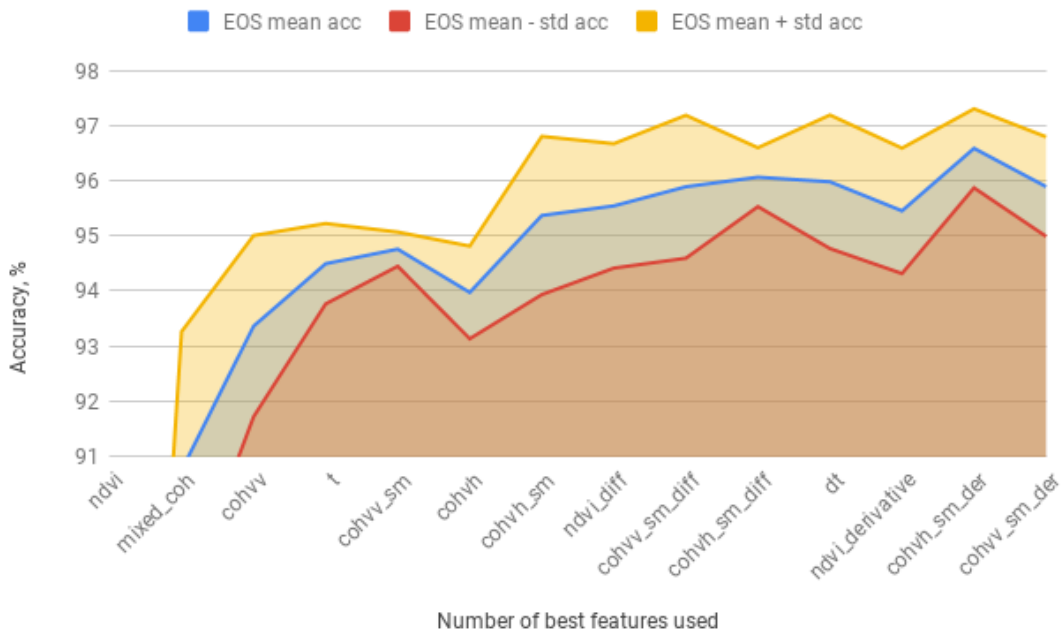

Figure 8. End of season accuracy for different number of best features

already gives more than 90% end of season accuracy. What is more, setup with 4 best features is only about 2% worse than the best setup with 13 features. Model that is constructed only on ndvi has end of season accuracy of 69.6%.

The event based accuracy for setups with different number of features is in Fig. 9. Surprisingly, two best features show the best performance. Overall, accuracy decreases with growing number of features.

Based on Fig. 8- 9, it could be seen that setup with 4 features has the best performance: second event-based score and approximately the same level as the best EOS accuracy. Also, a small number of features makes the model simpler and less resource consuming. These features (ndvi, mixed_coh, cohvv and t) will be used in further analysis.

4.3 Optimization process

Next step is to choose the best optimization settings, which includes choice of optimizer, its parameters and initial learning rate. We will iterate through 6 known optimizers following their default parameters (recommended in their original papers) and record monitored metrics values.

Fig. 10 contains EOS accuracy for different optimizers. It could be seen that Nesterov-accelerated Adaptive Moment Estimation (Nadam) optimizer has slightly better perfor-

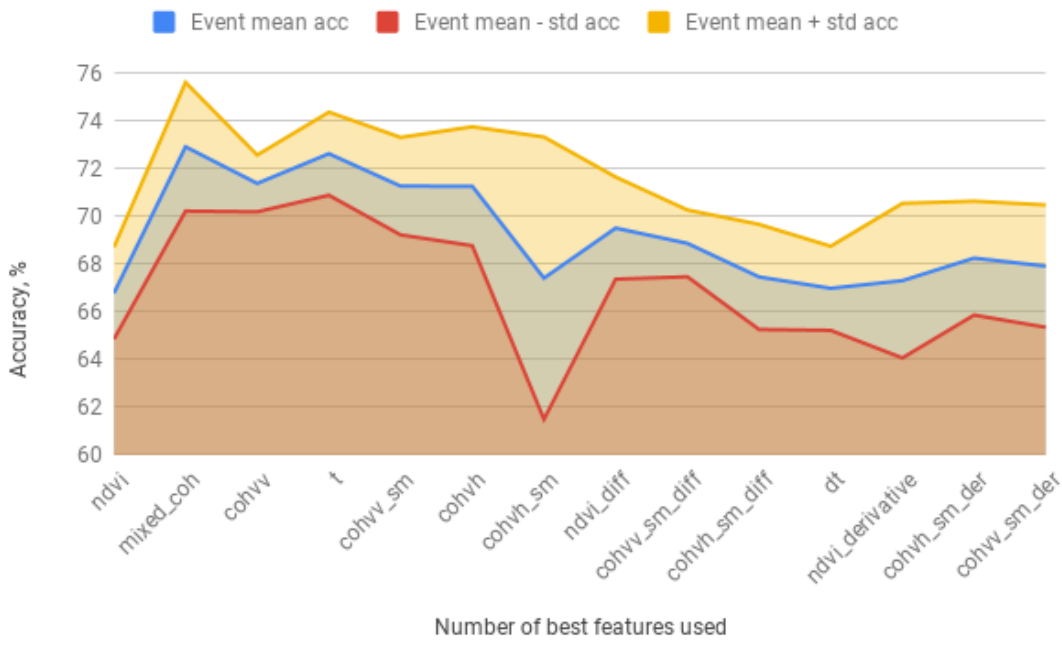

Figure 9. Event accuracy for different number of best features.

mance than other optimizers and significantly better than Adagrad.

Event performance is shown in Fig. 11. The rank of optimizers is completely different with EOS accuracy. Adagrad shows the highest accuracy while Adamax - the lowest. It is noticeable that Nadam is quite good in terms of event accuracy too - it divides second score with RMSprop.

Nadam optimizer shows the best performance based on two monitored metrics and it will be used for further analysis.

Nadam optimizer is recently developed algorithm [17], which includes Adaptive Moment Estimation (Adam) optimization and Nesterov accelerated gradient (NAG) [18]. While Adam [19] uses exponentially decaying average of past gradients (same as momentum) and their squares (Adadelta and RMSprop), NAG can be viewed as the correction for classical momentum method [20, 21] . Overall, Nadam combines several widely known optimizers and often outperforms them.

Nadam optimizer has following parameters:

- learning rate;
- beta1, beta2;
- schedule_decay;

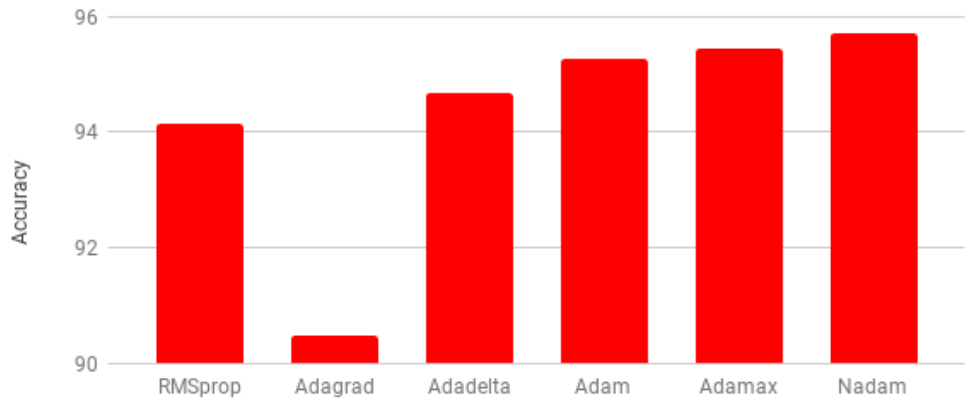

Figure 10. EOS accuracy for different optimizers.

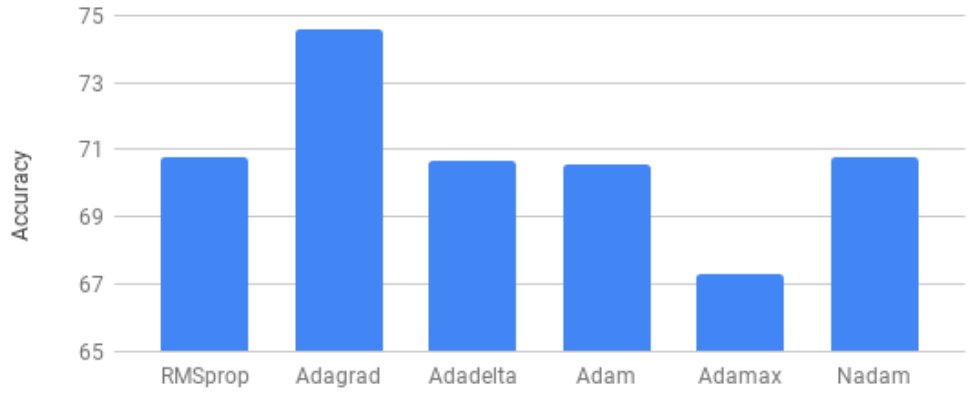

Figure 11. Event accuracy for different optimizers.

- epsilon;

Documentation of neural network framework Keras[3] suggests specific value for each parameter: lr=0.002, beta1=0.9, beta2=0.999, decay=0.004, epsilon=1e-07.

Epsilon, which is a fuzz factor for numerical computations, is not needed to tune. It is a small floating point constant (default value is 1e-07) used to avoid division by zero.

We performed experiments to choose best learning rate, by trying a value within order of magnitude from 0.00001 to 0.01.

It could be seen in Fig. 12 that order of magnitude 1e-4 for learning rate performs the best.

[3]https://keras.io/optimizers/

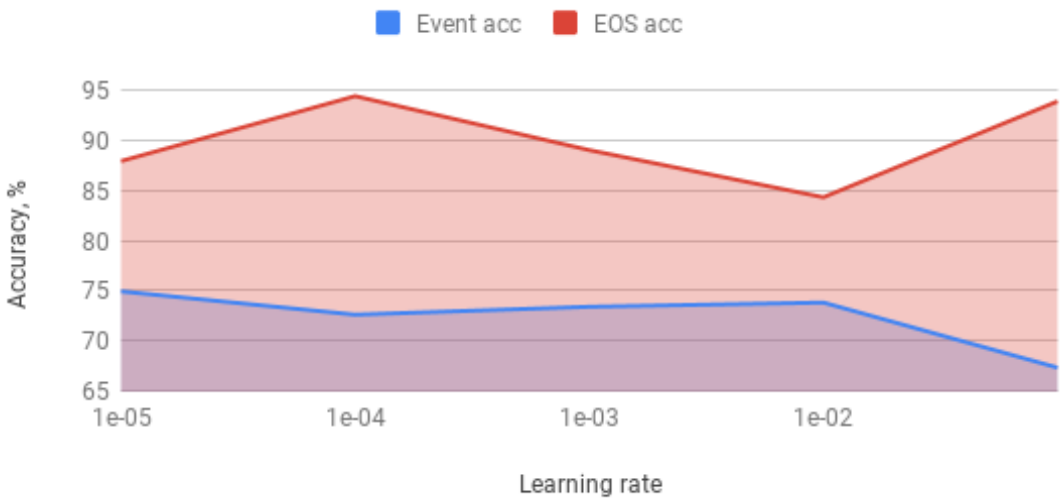

Figure 12. Performance within different initial learning rates.

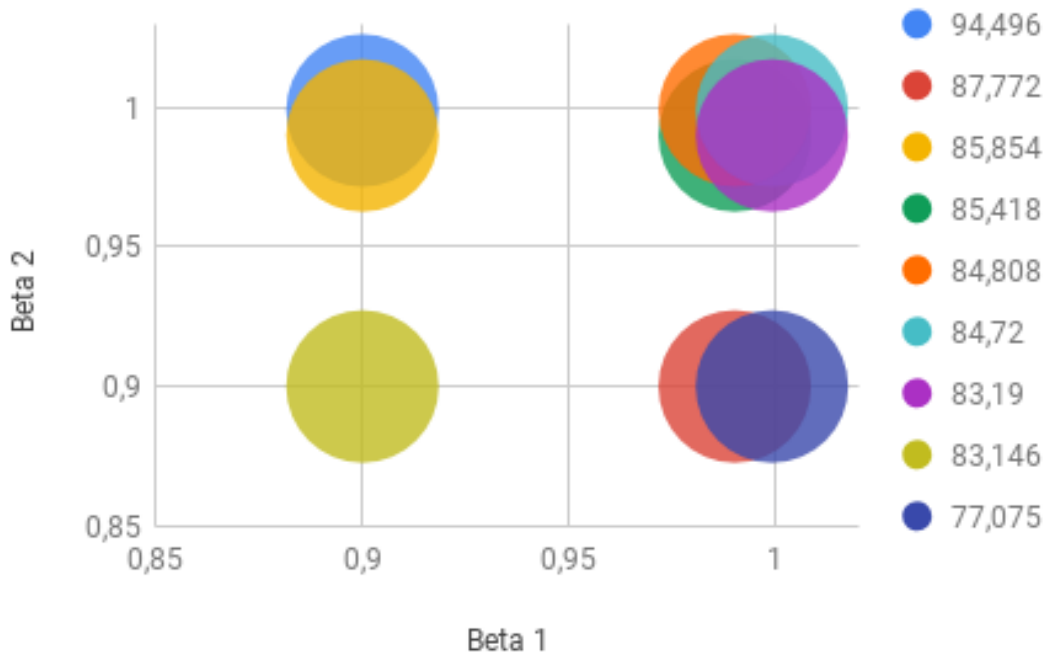

Figure 13. End of season performance within different beta.

To choose best values for beta1, beta2, we iterate through all possible combinations of (beta1, beta2), where each parameter can have a value among (0.9, 0.99, 0.999).

The end of season performance is on the Fig. 13.

It could be seen in Fig. 13 that EOS accuracy is significantly higher with default values of beta1 = 0.9 and beta2 = 0.999.

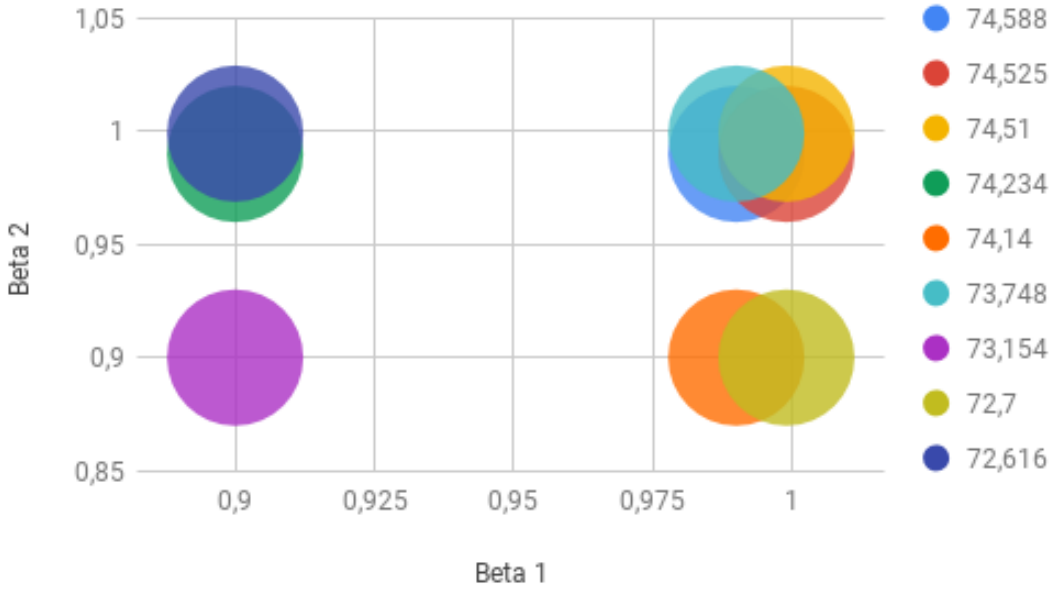

Figure 14. Event-based performance within different beta.

Event performance shown in Fig. 14 is approximately the same for different combinations of beta1 and beta2.

To conclude with, default values of beta1 and beta2 have the best performance.

The last parameter left not optimized is schedule decay. Schedule decay was chosen based on the best performance among values (0.002, 0.003, 0.004, 0.005).

Fig. 15 shows that default value of schedule decay 0.004 has the best performance.

In conclusion, the best configuration of optimizer and learning rate parameters is the following:

- Nadam optimizer;
- learning rate = 0.0008;
- beta1 = 0.9;
- beta2 = 0.999;

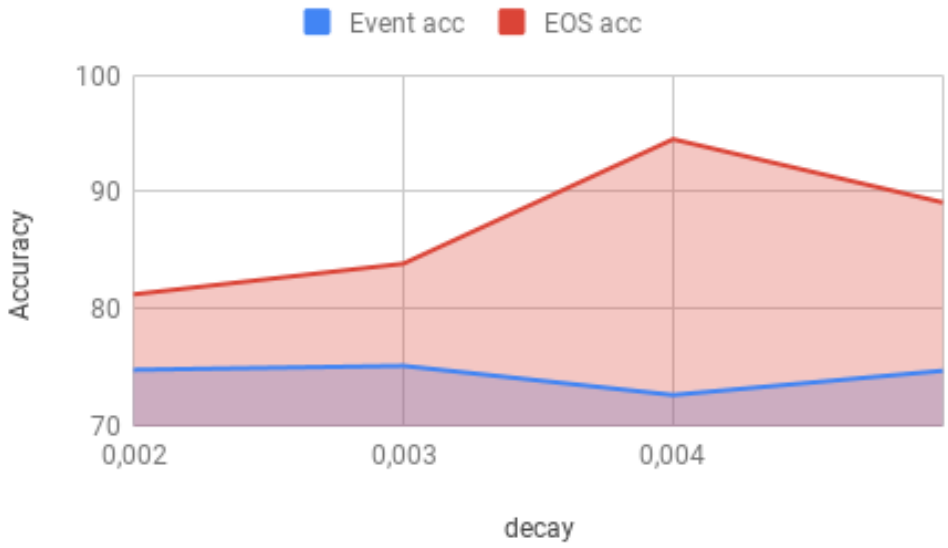

Figure 15. Performance with different schedule decay.

- epsilon = 1e-08;
- schedule decay = 0.004;

4.4 Architecture

There are a lot of different options how to create an architecture for neural network model. However, several strict limitations were mentioned previously: input tensor should have the shape [number_of_days, number_of_features] and output tensor: [number_of_days] and the architecture should be based on 1-D CNN.

Proposed architecture was described in General setup subsection. In this subsection, different hyperparameters will be tuned in order to obtain better monitored metric scores.

Architecture optimization steps:

1. **Number of layers.**
   The baseline architecture has 2 hidden 1-D CNN layers. Experiments with 1 and 3 hidden layers proves that 2 is still an optimal number given current dataset. It could be seen in Table 4 that performance is better with 2 hidden layers.

2. **Regularization.**
   There are different ways how to deal with overfitting in neural networks. Several

Table 4. Performance with different numbers of hidden layers.

Number of hidden layers	**Event accuracy, %**	**End of season accuracy, %**
1	69.2	91.1
2	**72.6**	**94.5**
3	61.1	89.4

possible options include Dropout and Batch Normalization. Dropout is less preferred technique to use with CNN, so Batch normalization performance will be studied.

The baseline architecture has two batch normalization layers: between first and second hidden CNN layer and between second hidden and output layer. The performance after disabling batch normalization layers is in Table 5.

Table 5. Performance with different combination of batch normalization layers.

Batch normalization layers	**Event accuracy, %**	**End of season accuracy, %**
Both layers	**72.6**	94.5
First layer is disabled	67.1	94.3
Second layer is disabled	63.7	**95.9**
Both layers are disabled	55.6	12.0

Table 5 shows that including both normalization layers is the best choice.

Batch normalization layer has several parameters that could be optimized. Main parameters to optimize are momentum (for the moving mean and variance) and boolean variables center (adding offset to tensor) and scale (if multiplication by γ is needed).

Default values for these parameters in Keras framework are 0.99 for momentum, scaling and centering are enabled. Results with different configurations of Batch normalization layer are listed in Table 6. The same configuration was applied for both layers.

It could be seen that tuning batch normalization parameters improves the performance. One configuration achieves the highest event accuracy 74%, other - highest EOS accuracy 96.6%. We will store these accuracies as the new baseline and will continue experiments with more balanced model that has 73.3% and 94.8% accuracies correspondingly.

3. **Number of filters and kernel size.** Most important parameters of CNN layers are number of filters and kernel size. Parameter filters determine the dimensionality

Table 6. Performance with different batch normalization parameters.

Momentum	Scale	Center	Event accuracy, %	EOS accuracy, %
0.99	TRUE	TRUE	72.6	94.5
0.9	TRUE	TRUE	70.6	94.7
0.999	TRUE	TRUE	64.5	94.8
0.9	TRUE	FALSE	55.6	96.4
0.99	TRUE	FALSE	65.3	94.5
0.999	TRUE	FALSE	67.0	94.0
0.9	FALSE	TRUE	**74.0**	93.7
0.99	FALSE	TRUE	**73.3**	**94.8**
0.999	FALSE	TRUE	60.9	**96.6**

of an output tensor. Kernel size is the length of the 1D convolution window. Performed experiments with different number of filters and kernel size are shown in Table 7. The highest achieved EOS accuracy is 96.5%, event accuracy - 76.1%. It could be seen that configuration with smaller number of filters and kernel size has generally higher EOS accuracy but lower event accuracy. For the larger number of filters and kernel size, event accuracy is much better, while EOS accuracy - worse.

Table 7. Kernel size and number of filters.

Filter1	**Kernel 1**	**Filter2**	**Kernel 2**	**Event accuracy, %**	**EOS accuracy, %**
10	10	10	10	71.4	93.9
10	10	20	10	50.8	94.3
20	10	20	10	50.8	**96.5**
20	10	40	10	68.0	94.8
20	20	40	10	75.3	91.8
20	20	40	20	**76.1**	89.2
30	10	10	10	74.2	87.3
30	10	20	10	75.3	89.8
30	10	30	10	73.3	91.7
30	20	30	20	74.1	90.8
40	10	10	10	74.3	88.2
40	10	20	10	75.0	90.2
40	10	30	10	74.2	92.6
40	10	40	10	74.8	90.4
40	20	40	20	74.9	88.2
40	30	40	30	75.4	91.5

Trained filters (weights of CNN layer) could be visualized. Ideally, they should somehow correlate with signature of mowing event (ndvi decrease and coherence increase). Examples of trained filters are on the Fig. 16.

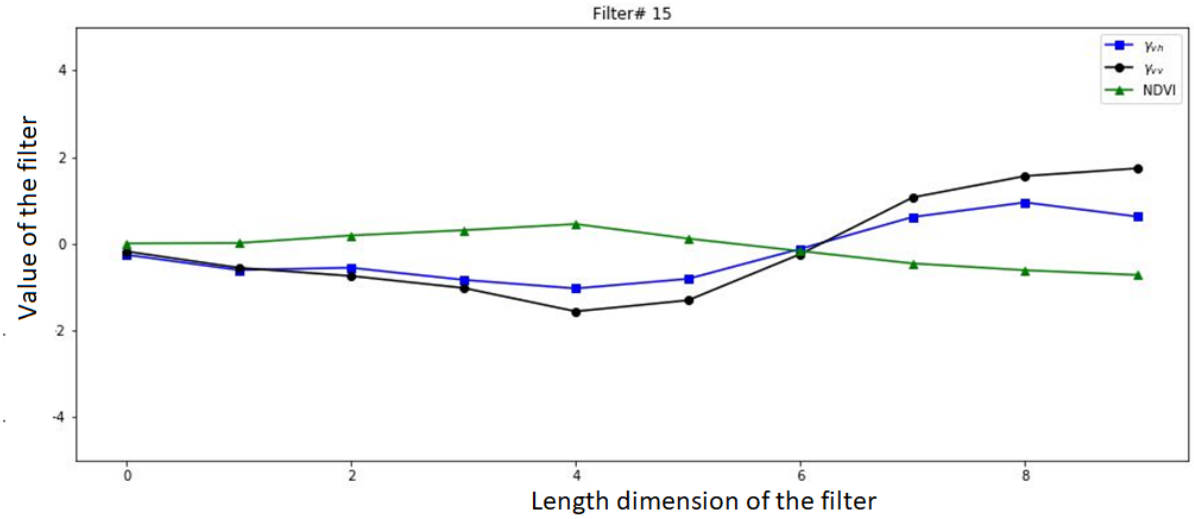

Figure 16. Example of trained filter.

The X-axis represents the length dimension of a filter, y-axis - value of the filter in correspondent point. Green color stands for ndvi dimension of a filter, black and blue - for cohvv and cohvh respectively. The given example shows how filters are trained to replicate signature of mowing event. If the number of filters is too big, some of the filters are becoming flat.

4. **Activation functions.** Activation function is also a parameter that could be optimized. The model includes three CNN layers; each of them can have own activation function. However, activation function of the last layer is fixed to sigmoid, as the output of the network should be probability of mowing event. Experiments with different activation functions that were assigned to first and second layers only are listed in the Fig. 17. Experiments show that linear activation has the worst performance, while sigmoid and softmax have highest accuracies. The average event accuracy for sigmoid activation is 72.6%, for softmax activation - 72.3%. The average end of season accuracy is 94.5% for sigmoid but 95.1% for softmax. Overall, it could be concluded that softmax activation is slightly better.

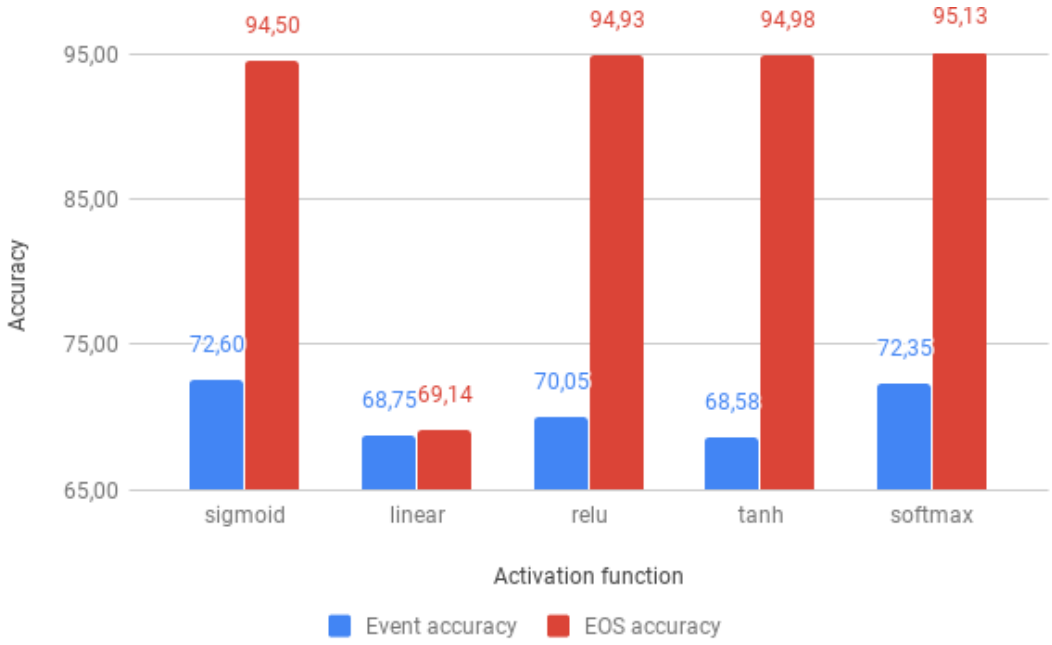

Figure 17. Performance with different activation functions used in hidden layers.

4.5 Summary of the chapter

In this chapter, the initial setup of the model was described. The model is based on many-to-many 1D CNN architecture. Two evaluation metrics are used: event accuracy and end of season accuracy. Two baseline scores for the model performance were listed (best of them are 69.4% for event accuracy and 96.0% for the end of season accuracy). In the feature engineering section, we ranked features by their importance and selected an optimal subset of 4 features. After that, the optimal optimizer was chosen and its parameters were tuned. Finally, the amount and parameters of convolutional and regularization layers were tuned. Several configurations were found that outperform baseline accuracies. The highest found event accuracy was 76.1% while 96.6% EOS accuracy was the best. The most balanced setup (highest average value of both accuracies) reaches 73.3% and 94.8% of event and EOS accuracy respectively.

5 Model generalization

This chapter describes experiments towards creation of universal model (disregards area and season). The main idea of the chapter is an easy adaptation model for conditions with a small number of labeled events. The last section is about the reject region techniques. They allow selecting only highly confident predictions, which is very important for decision-making processes (e.g. sanctioning).

5.1 Transfer learning

Transfer learning is a research problem of reusing the knowledge obtained from solving one problem for solving another (known in the scientific community as domain and task respectively) [22]. In terms of mowing events detection, it could be viewed as applying model, trained on fields from a specific area in a specific season, to predict events at any field in any season.

Transfer learning is a hot topic in the neural network community nowadays. Applications in computer vision domain are widely known. As a lot of computer vision systems are based on CNN, we could adapt ideas from this field for the current task.

First convolutional layers of CNN architecture are responsible for low-level features descriptors, medium ones - for extraction high-level features, while only the last ones are applying all of these features for solving given task. It means that first layers of the model, trained on a big dataset, could be used for solving the specific task from a similar domain, where only a small dataset is available. There are two general ways of how trained weights could be used in other tasks. The first one is when the transferred weights are fixed and are not changed during training. In second way, domain weights are used only for initialization. The weights are changed during training, but the starting values are nearly optimal. The second method is called fine-tuning.

5.1.1 Transfer knowledge to another country

To test transfer learning capabilities for mowing events detection, a big dataset from Swedish and Danish fields in 2018 was formed (4000 fields in total). These fields were manually labeled by human experts. The trained weights will be adapted for Estonian fields in 2018.

Large dataset requires a big number of features. Two setups with 8 and 11 best features, described in Features engineering subsection, will be used for the current task. Both fine-tuning and fixing weights will be applied.

The results of performed experiments are listed in Table 8. It could be seen that the performance remains at the same level. However, configuration with 8 features and transferred weights from the first layer achieves high event accuracy **75.1%**.

Table 8. Transfer to different country.

Method	Features	Fixed first layers	Event acc., %	EOS acc., %
Fine-tuning	8	0	69.6	95.0
Fixed weights	8	1	**75.1**	91.0
Fixed weights	8	2	61.6	65.1
Fine-tuning	11	0	69.2	**95.4**
Fixed weights	11	1	73.1	95.0
Fixed weights	11	2	59.1	66.4

5.1.2 Transfer knowledge to another season

Experiments described in the previous section are based on the field data from the same year (2018). The aim of this section is to improve the performance of the Estonian 2018 model by using Estonian 2017 data. The experiments setup is the same as in the case of transferring across countries.

Table 9. Transfer to different season.

Method	Features	Fixed first layers	Event acc., %	EOS acc., %
Fine-tuning	8	0	74.9	91.0
Fixed weights	8	1	74.6	89.7
Fixed weights	8	2	66.2	59.0
Fine-tuning	11	0	74.5	91.9
Fixed weights	11	1	**75.5**	**92.1**
Fixed weights	11	2	64.0	56.8
Baseline scores	-	-	69.4	96.0
Best Chapter 4 scores	4	-	76.1	96.6

It could be viewed from Table 9 that performance is on a similar level with transferring to other countries. Event accuracy is slightly higher, while EOS accuracy is generally lower. Setup with 11 features and fixed first CNN layer reaches 75,5% event accuracy, which outperforms most of the configurations from section Model construction.

To conclude with, transfer learning is a promising technique that could improve model performance. Obtained knowledge from other countries and other season data helps to outperform baseline setup in several configurations. However, transfer learning is generally applied, when domain dataset is much larger (to the orders of magnitude). In performed experiments, combined Swedish and Danish dataset (used in transferring to other countries section) is about 3 times larger than Estonian 2018 (used for testing results), Estonian 2017 (transferring to another season) is even smaller. Usage of the larger dataset could potentially boost model performance.

5.2 Reject region

Sometimes the model is not confident enough to give a reliable decision about the state of the field. It does not always mean that the model is not good enough. Given inaccurate, incomplete or uncertain data we cannot expect reliable and confident prediction. It is better to allow model not to give a prediction if it is not confident enough. In this way, obtained predictions will be more accurate, while rejected fields could be double-checked with human experts.

The set of rejected fields forms the reject region. Given desired EOS precision and recall of mowing detection for non-rejected fields, the necessary reject region can be constructed using an algorithm described below.

CNN predictions are in the range from zero to one. Predictions that are in the middle of the interval are naturally less confident. The reject region using CNN outputs could be defined as an interval (t_{low}, t_{upper}), where $0 < t_{low} < t_{upper} < 1$. It could be noticed that the precision is not affected by t_{low} while t_{upper} is fixed (because t_{low} only determines the number of true negatives and false negatives). An algorithm for finding precise values of t_{low} and t_{upper} is the following:

1. Choose maximum t_{low} that satisfy necessary recall on validation data.
2. Choose t_{upper} that satisfies precision requirement.

The main assumption behind this algorithm is that the model is well calibrated. Predicted probabilities of the well-calibrated model correspond to the real probabilities: out of 100 fields, predicted as mown with probability 80%, around 80 should be actually mown.

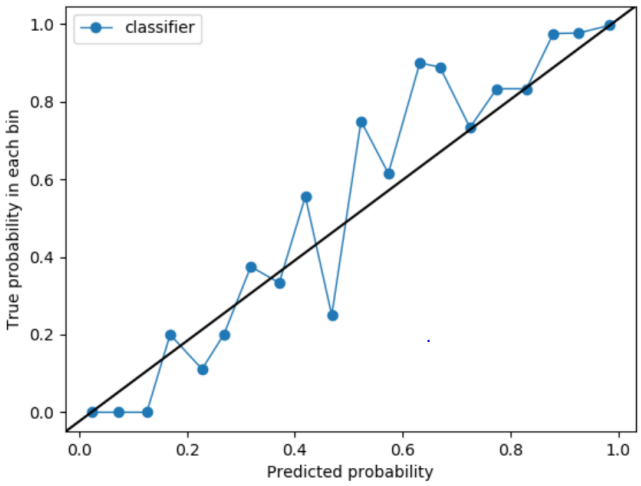

Figure 18. Calibration plot.

We will check if the model needs further calibration by studying the calibration plot. The X-axis of such a plot is a range from zero to one, divided by N (N=20 will be used) even bins. Each bin represents an interval for predicted probability. The Y-axis will show the fraction of not mown among fields that belong to a specific bin. The perfectly calibrated classifier has the curve close to diagonal.

The calibration plot of the model is in Fig. 18. It could be noticed that the classifier's curve points are almost always near the diagonal, with no significant bias under or above the diagonal. It means that the model is quite well calibrated and no additional calibration techniques are needed.

Performance of the reject region algorithm will be tested with three configurations of desired precision and recall values:

1. Precision = 90%; recall = 90%.
2. Precision = 97%; recall = 75%.
3. Precision = 75%; recall = 97%.

Reject region borders were found on a validation set with the usage of desired precision and recall. The performance of the obtained reject region was evaluated on the test set. In this section, the main focus is on end of season mowing detection. Besides accuracy, other evaluation metrics will be used.

Baseline setup consists of CNN-based architecture, described in section Model construction. It does not use any reject region techniques.

Given the default probability threshold of 50%, baseline setup has the following performance characteristics:

- Accuracy 92.4%.
- True positive rate (TPR) equals to $\frac{TP}{TP+FN} = 96.3\%$.
- True negative rate (TNR) equals to $\frac{TN}{TN+FP} = 67.1\%$.
- Positive predictive value (PPV) equals to $\frac{TP}{TP+FP} = 95.0\%$.
- Area under the ROC Curve (AUC) is 94%.

Precision recall plot is in Fig. 19.

For the first experiment, we set the desired precision to 90% and recall to 90% and run algorithm described above.

Obtained reject region is all samples that have the maximum prediction during the season in the range from 0.04 to 0.789. The region is quite big, around 25% of data will be rejected to predict. However, some of the evaluation metrics values were improved.

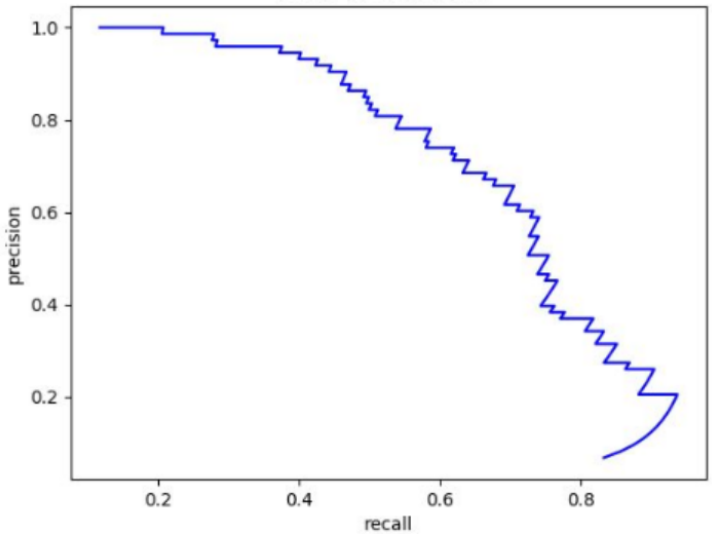

Figure 19. Precision recall plot for baseline setup.

- Accuracy 86.5%.
- TPR is 85.6%.
- TNR is 92.7%.
- PPV is 98.7%.
- AUC score is is 99%.

The precision-recall plot for this setup is in Fig. 20. It could be seen that the plot is much better than the baseline one: the area under the curve is significantly larger. TNR, PPV, and AUC are much better for this setup, while TPR and overall accuracy are worse.

Setup with desired precision 75% and recall 97% has the following performance: Reject region is from 0.13 to 0.86.

- Accuracy 80.7%.
- TPR is 78.5%.
- TNR is 95.1%.
- PPV is 99.1%.
- AUC score is is 96%.

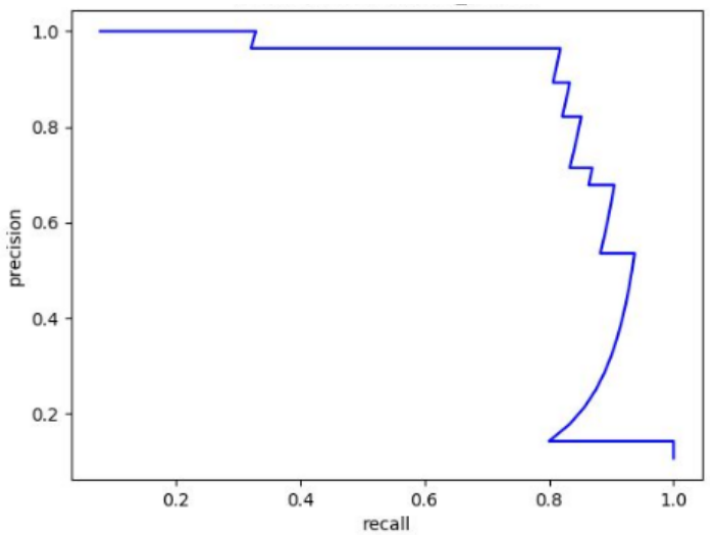

Figure 20. Precision recall plot for desired precision and recall of 90%.

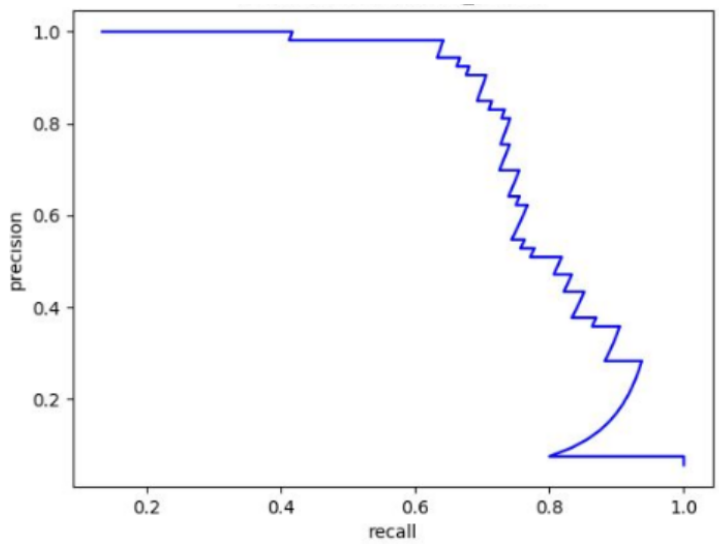

Figure 21. Precision recall plot for desired precision 75% and recall 97%.

It could be seen in Fig. 21 that precision-recall plot is better than baseline but worse than in the previous experiment. We received very high PPV 99.1% and TNR 95.1%, while other metrics were decreased.

Setup with desired precision 97% and recall 75% has the reject region in the range from 0.039 to 0.519. Evaluation metrics are generally similar to the baseline ones:

- Accuracy 92.4%.
- TPR is 95.9%.
- TNR is 69.5%.
- PPV is 95.4%.
- AUC score is is 97%.

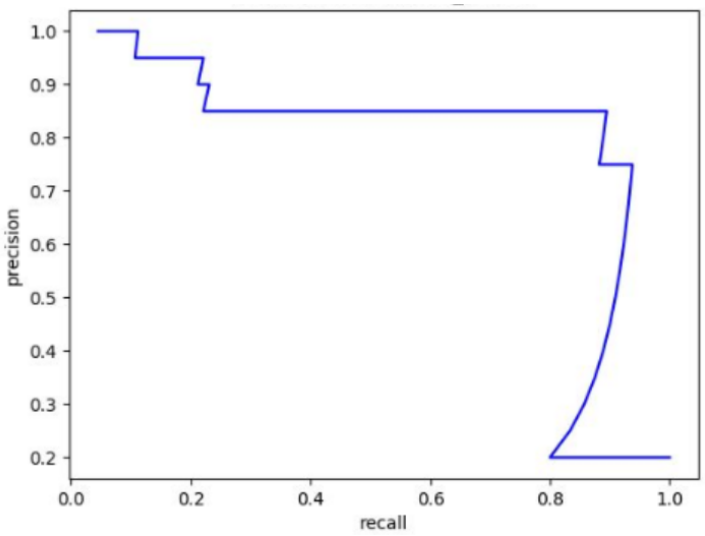

Figure 22. Precision recall plot for desired precision 97% and recall 75%.

The recall precision plot is in Fig. 22. Overall, it is close to the baseline plot.

Table 10. Reject region summary performance.

Recall th	**Precision th**	**TPR**	**TNR**	**PPV**	**ACC**	**AUC**	**Prob. range**	**Rejected, %**
-	-	**96.3**	67.1	95.0	**92.4**	94	-	0
90	90	85.6	92.7	98.7	86.5	**99**	(0.04, 0.78)	24.8
97	75	78.5	**95.1**	**99.1**	80.7	96	(0.13, 0.86)	30.0
75	97	95.9	69.5	95.4	**92.4**	97	(0.04, 0.52)	12.8

Summary of all experiments within subsection Reject region is in Table 10. It could be seen that the fraction of rejected fields is big and could be viewed as unsatisfactory. However, obtained performance is very high: 0.99 AUC, 99.1% PPV, 95.1% TNR, 95.9% TPR, 92.4% for different configurations depending on which metric is the priority. There

is a trade-off between fraction of rejected fields and performance improvement. By setting realistic desired recall and precision, we can obtain lower number of rejected fields with still better evaluation metrics values.

5.3 Summary of the chapter

Experiments with transfer learning techniques were performed on relatively small domain datasets: 4000 fields for joined Danish and Swedish 2018 and 1500 fields for Estonian 2017 data, while the target dataset, Estonian 2018, is about 2000 fields. However, the results are approximately at the level described in the section Model construction, which makes these methods promising for usage in situations with larger domain datasets.

The reject region experiments showed that unrealistically high desired precision and recall will result in a very big rejected region. Depending on the needs, we could vary these parameters to achieve either very high TPR (95.9% was obtained) or TNR and PPV (95.1% and 99.1% were received respectively). The method is very flexible, which allows adapting the model to different evaluation requirements after proper tuning.

6 Conclusion

The main goal of the thesis was to build a reliable grassland mowing event detection system based on Sentinel-1 and -2 imagery time series.

In the processing step, ndvi outlier detection method was proposed. In the feature engineering step, new features were derived (using smoothing, differences, derivatives and cross-feature dependencies) and their importance, together with base features. An optimal subset of features was chosen, which includes ndvi, a quadratic mean of coherence in VH and VV polarization, time and coherence in VV polarization.

In Model construction section, evaluation metrics were introduced (event accuracy and end of season accuracy) together with baseline scores. According to evaluation metrics results on target Estonian 2018 dataset, optimal architecture was constructed, which included a number of layers, their types, parameters, activation functions, choice of optimizer, its parameters and learning rate. The highest event accuracy obtained was 76.1% (the baseline score was 69.4%), the highest end of season accuracy was 96.6% (while the baseline was 96.0%). More balanced derived setup achieved 73.3% event and 94.8% end of season accuracy.

In the transfer learning section, trained weights on Swedish and Danish 2018 and Estonian 2017 data were obtained. The weights were transferred to Estonian 2018 model in order to improve its performance. Two approaches: usage of transferring weights only for initialization and weights fixation during training were studied. Performance remained approximately on the same level, which could be explained with a relatively small size of source datasets.

Reject region methods allow to obtain high performance on a specific subset of the dataset (where the classifier is confident enough), while other parts of the dataset are not predicted. An algorithm was proposed that finds reject region based on desired precision and recall. Highest achieved true positive rate was 95.9%, true negative rate 95.1%, positive predictive value 99.1%, accuracy 92.4% and area under the ROC curve 99% (obtained from different configurations). However, for these setups 50 to 70% of the data were rejected.

Future work will be possible after new datasets will be obtained. The main areas of further development are transfer learning and reject region techniques. Also, with a larger amount of data, model architecture could be revised: e.g. additional hidden layers might be added.

References

[1] S. Voigt, F. Giulio-Tonolo, J. Lyons, J. Kučera, B. Jones, T. Schneiderhan, G. Platzeck, K. Kaku, M. K. Hazarika, L. Czaran, *et al.*, "Global trends in satellite-based emergency mapping," *Science*, vol. 353, no. 6296, pp. 247–252, 2016.

[2] A. Lefebvre, C. Sannier, and T. Corpetti, "Monitoring urban areas with Sentinel-2a data: Application to the update of the copernicus high resolution layer imperviousness degree," *Remote Sensing*, vol. 8, no. 7, p. 606, 2016.

[3] K. von Schuckmann, P.-Y. Le Traon, N. Smith, A. Pascual, P. Brasseur, K. Fennel, S. Djavidnia, S. Aaboe, E. A. Fanjul, E. Autret, *et al.*, "Copernicus marine service ocean state report," *Journal of Operational Oceanography*, vol. 11, no. sup1, pp. S1–S142, 2018.

[4] J. Rouse Jr, R. Haas, J. Schell, and D. Deering, "Monitoring vegetation systems in the great plains with erts," 1974.

[5] T. Tamm, K. Zalite, K. Voormansik, and L. Talgre, "Relating Sentinel-1 interferometric coherence to mowing events on grasslands," *Remote Sensing*, vol. 8, no. 10, p. 802, 2016.

[6] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[7] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *International Conference on Web-Age Information Management*, pp. 298–310, Springer, 2014.

[8] N. Kussul, G. Lemoine, F. J. Gallego, S. V. Skakun, M. Lavreniuk, and A. Y. Shelestov, "Parcel-based crop classification in Ukraine using Landsat-8 data and Sentinel-1a data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 6, pp. 2500–2508, 2016.

[9] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[10] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Urban change detection for multispectral earth observation using convolutional neural networks," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2115–2118, IEEE, 2018.

[11] S. Vafaei, J. Soosani, K. Adeli, H. Fadaei, H. Naghavi, T. Pham, and D. Tien Bui, "Improving accuracy estimation of forest aboveground biomass based on incorporation of alos-2 palsar-2 and sentinel-2a imagery and machine learning: A case study of the hyrcanian forest area (iran)," *Remote Sensing*, vol. 10, no. 2, p. 172, 2018.

[12] M. Kang, K. Ji, X. Leng, and Z. Lin, "Contextual region-based convolutional neural network with multilayer fusion for sar ship detection," *Remote Sensing*, vol. 9, no. 8, p. 860, 2017.

[13] M. G. Castro Gomez, "Joint use of sentinel-1 and sentinel-2 for land cover classification: A machine learning approach," *Lund University GEM thesis series*, 2017.

[14] M. D. Johnson, W. W. Hsieh, A. J. Cannon, A. Davidson, and F. Bédard, "Crop yield forecasting on the canadian prairies by remotely sensed vegetation indices and machine learning methods," *Agricultural and forest meteorology*, vol. 218, pp. 74–84, 2016.

[15] N. Kolecka, C. Ginzler, R. Pazur, B. Price, and P. Verburg, "Regional scale mapping of grassland mowing frequency with Sentinel-2 time series," *Remote Sensing*, vol. 10, no. 8, p. 1221, 2018.

[16] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[17] T. Dozat, "Incorporating nesterov momentum into adam." ICLR 2016 workshop submission, 2016. https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ.

[18] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence o(1/k2)," vol. 269, p. 543– 547, 1983.

[19] B. J. L. Kingma, D. P., "Adam: a method for stochastic optimization.." International Conference on Learning Representations, 2015.

[20] S. Ruder, "An overview of gradient descent optimization algorithms," 2017. http://ruder.io/optimizing-gradient-descent/, Last visited on: 2019-05-16.

[21] R. Gylberthr, "Momentum method and nesterov accelerated gradient," 2018. https://medium.com/konvergen/momentum-method-and-nesterov-accelerated-gradient-487ba776c987, Last visited on 2019-05-16.

[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Viacheslav Komisarenko**,

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,
   Farming events detection from Sentinel-1 and -2 satellite imagery time series with deep learning,
   supervised by Sherif Sakr, Kaupo Voormansik and Yousef Essam.

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.

Viacheslav Komisarenko, 20.05.2019