Computers and Electronics in Agriculture 163 (2019) 104859

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Crop yield prediction with deep convolutional neural networks

Petteri Nevavuori[b,*], Nathaniel Narra[a], Tarmo Lipping[a]

[a] Tampere University of Technology, Finland
[b] Mtech Digital Solutions Oy, Finland

ARTICLE INFO

Keywords:
Crop yield prediction
Convolutional neural network
Wheat
Barley
UAV
Multispectral
NDVI
Growth phase

ABSTRACT

Using remote sensing and UAVs in smart farming is gaining momentum worldwide. The main objectives are crop and weed detection, biomass evaluation and yield prediction. Evaluating machine learning methods for remote sensing based yield prediction requires availability of yield mapping devices, which are still not very common among farmers. In this study Convolutional Neural Networks (CNNs) – a deep learning methodology showing outstanding performance in image classification tasks – are applied to build a model for crop yield prediction based on NDVI and RGB data acquired from UAVs. The effect of various aspects of the CNN such as selection of the training algorithm, depth of the network, regularization strategy, and tuning of the hyperparameters on the prediction efficiency are tested. Using the Adadelta training algorithm, L^2 regularization with early stopping and a CNN with 6 convolutional layers, mean absolute error (MAE) in yield prediction of 484.3 kg/ha and mean absolute percentage error (MAPE) of 8.8% was achieved for data acquired during the early period of the growth season (i.e., in June of 2017, growth phase < 25%) with RGB data. When using data acquired later in July and August of 2017 (growth phase > 25%), MAE of 624.3 kg/ha (MAPE: 12.6%) was obtained. Significantly, the CNN architecture performed better with RGB data than the NDVI data.

1. Introduction

Development-minded farmers have practiced what is now known as precision agriculture long before the dawn of the computing age. They were able to deduce sources of field variability and the actions to take for trying to secure an enhanced level of crop yields. The farmers accomplished this by taking notes of their fields during growing seasons and harvest time operations and tried to figure out the best actions for the year to come based on the accumulated knowledge and experience. However, as studied by Wolfert et al. (2017), the increase in data-producing devices and sensors has been an on-going trend in agriculture having enabled the farmers to shift towards data-driven decision-making. This is commonly called smart farming. A comprehensive review of various objectives and techniques used in smart farming can be found in Kamilaris et al. (2017).

An important trend in smart farming is the use of remote sensing to facilitate the extraction of information relevant for data-driven decisions (Miyoshi et al., 2017; Matikainen et al., 2017). Remote sensing data can be acquired from satellites such as ESA's Sentinel-2A, for example. The problem with the satellite data is that if there is a cloud cover during the overflight of the satellite, no useful data are obtained. The spatial resolution of Sentinel imagery is at best 10 m, which is enough for many applications but too low to allow using texture-based information in the images. Satellite data contains predefined wavelength bands from both the visible and the Near Infrared (NIR) spectral regions. In satellite-borne sensors, designed keeping in mind agricultural applications, the spectral bands are optimized for the calculation of relevant indices such as the Normalized Difference Vegetation Index (NDVI), for example. The spatial and temporal resolution of satellite data will improve in years to come, however, cloud cover will remain an obstacle, especially in northern climate.

Using Unmanned Aerial Vehicles (UAVs), or drones, for data acquisition offers better spatial resolution, the data acquisition time can be selected by the user and the data can be acquired also in cloudy conditions. Spectral wavelengths can be selected by using appropriate camera; UAV-mountable RGB-NIR cameras are available at affordable price. The drawback is that the UAV has to be operated locally and managing the data and extracting relevant information requires highly specialized skills. As the variety of UAVs and UAV-mountable sensors is high compared to satellite-borne sensors, analysis frameworks and services based on UAV-borne data are not yet equally developed. In Näsi et al. (2017), extraction of information related to the biomass and nitrogen content of vegetation (barley and grass) in test fields using various modalities of remote sensing data (satellite/aircraft/drone using RGB/multispectral/hyperspectral sensors) has been considered.

Information relevant for decision making in agriculture can be

* Corresponding author.
E-mail addresses: petteri.nevavuori@mtech.fi (P. Nevavuori), nathaniel.narra@tuni.fi (N. Narra), tarmo.lipping@tuni.fi (T. Lipping).

https://doi.org/10.1016/j.compag.2019.104859
Received 17 April 2019; Accepted 13 June 2019

extracted from remote sensing data by means of machine learning. Traditional machine learning techniques involve feature extraction as an initial stage. Based on the features, different tasks such as crop classification, weed detection or yield prediction can be addressed. In Ruß (2009) several traditional machine learning techniques have been applied to the task of yield prediction. It is, however, often difficult to find optimal features and the ability of the traditional methods to learn from the data is limited. With advancements in computational technology, the development and training of novel multilayer algorithms has become feasible. These methods are commonly referred to as deep learning. Among the various deep learning paradigms, Convolutional Neural Networks (CNNs) have proved especially efficient in image classification and analysis. In case of CNNs no features need to be precalculated as the feature extraction operation is performed by the convolutional layers of the network and optimal features are obtained in the course of training. Due to this kind of structure, CNNs require large amounts of training data to converge. The advantage of CNNs compared to traditional machine learning methods in crop yield prediction is discussed, for example, in You et al. (2017). CNNs have been successfully applied to crop classification (Chunjing et al., 2017) and weed detection (Sa et al., 2017; Milioto et al., 2017).

In working towards an effective in-season crop yield predictor model for the northern climate, our effort in this preliminary study is to develop a CNN based deep learning framework using UAV-acquired multispectral data. RGB and NDVI images, representing patches of wheat and barley fields, are fed as input data to a CNN and training is performed to tune the network parameters. In addition to testing the usefulness of deep learning models for crop yield prediction in general, we also experiment with various setups and training schemes of the CNN model. Training a deep learning network is typically an iterative process as there is a substantial number of cross-related parameters to tune. We first select the most promising training algorithm from three candidates (see Section 3.1) and determine the optimal number of convolutional layers of the CNN. After that, we optimize the performance of the network in terms of regularization and parameters of the training algorithm. The optimized framework is evaluated using two types of input data (RGB and NDVI) and three patch sizes (10, 20 and 40 m).

2. Data and methods

2.1. Data acquisition

The nine crop fields selected for this study are located in the vicinity of the city of Pori (61°29′6.5″N, 21°47′50.7″E). The total area of the fields was approximately 90 ha. The main crops grown in the fields were wheat and malting barley, however the model was trained over the fields without making a distinction between the crop type.

Multispectral data were acquired from these fields during the growing season of 2017 (i.e., from June to August; see Table 1). The data were collected with a single Airinov Solo 3DR UAV equipped with Parrot's NIR-capable SEQUIOA-sensor. The images of individual spectral bands were stitched together to form complete orthogonal RGB and NDVI rasters of distinct fields using the Pix4D software.

The UAV data were organized into two sets according to the time of data acquisition to see if the phase of the growing season had an effect on predicting the yield from the input image. Growing phase here is defined as the percentage of total thermal time on the day of imaging. Thermal time for each day was calculated as the magnitude of daily average temperature above 5 °C. The temperature readings were downloaded from the Finnish Meteorological Institute. Beginning of July 2017 was chosen as the separating time point between the two data sets as the UAV data dispersed equally enough around that date. The data sets containing images only prior to July 2017 were labeled as *early* (growth phase < 25% of the total thermal time) and the remaining data as *late* (growth phase > 25% of the total thermal time). Details of the fields, crops, imaging dates and corresponding growth phases are listed in Table 1.

The field-wise image data were then processed using a sliding window to extract geolocationally matched pairs of input image frames (UAV data) and targets (yield data) of predefined size from all the fields. The step of the applied sliding window was chosen to be 10 m according to the resolution of Sentinel-2A satellite data considering the possibility of using satellite data as an additional input to the network in future studies. Image frames of sizes 10 × 10 m, 20 × 20 m and 40 × 40 m were considered. The resolution of the UAV data was 0.3125 m or 32 pixels per 10 m. The overall number of extracted frames according to crop fields is given in Table 2. The individual data frames were treated as independent inputs fed to the CNN models. The process of data preparation prior to and during training is illustrated in Fig. 1.

The harvest yield data was acquired during September 2017 using two distinct setups attached to the harvesters: Trimble CFX 750 and John Deere Greenstar 1. As the yield measurement devices produce an irregular set of data points with multiple attributes, the data had to be processed to be handled as rasters of field-wise yield from the viewpoint of the trainable network. The data points were first filtered according to (Tiusanen, 2017) to preserve only points corresponding to harvester speed between 2 and 7 km/h and yield between 1500 and 15,000 kg/ha. The filtering and generation of rasterizable vector files was done using the FarmWorks software. The field-wise vector data files were then rasterized by interpolating them using an exponential point-wise inverse distance algorithm. Yield values constitute targets the model is trying to predict during the training of the CNNs. Thus, yield values were also extracted using sliding windows similar to the UAV images to have geolocationally matching pairs of inputs and targets. Yield values were then averaged over the analysis window to obtain scalar target values. The histograms and statistics of yield values for point data as well as window-averaged data using three sample area window sizes (10 m, 20 m and 40 m) over all crop fields are given in Fig. 2. As can be expected, the larger the window, the more concentrated the yield values are around the mean.

For clarity, we also visualize several NDVI and RGB input images of the largest sample area window size (40 m) with their corresponding yields in Fig. 3 with the color bar corresponding to yield image value range. The images with similar identifiers are from the same location. However, the target for the network will be the mean of the yield values over the analysis window corresponding to the input area. It is also important to note that the network was trained separately for RGB and NDVI input images so that the possible misalignment between the two image sources does not affect prediction results. This kind of approach enables us to evaluate which one of the two input sources, RGB or NDVI, gives better prediction results.

2.2. Building the convolutional neural network

Convolutional neural networks, or CNNs, are deep learning models specialized in handling grid-like data. Such data can be images or rows of multi-column data. *Deep learning* refers to models composed of multiple layers. Generally, a model is viewed as deep if it has at least an input layer, one hidden layer and an output layer. The term *neural* on the other hand refers to the fact that originally the operation principle of artificial neural networks was taken from that of the brain, containing neurons as its basic building blocks. Compared to traditional feedforward neural networks, CNNs possess some special features making them extremely efficient in finding salient features within the data. Some of these features are:

1. exploitation of the convolution operation
2. post-convolution pooling
3. specific non-linear activation functions.

In the following we provide a brief description of these elements

Table 1
Details of crops and their varieties sown in each of the 9 fields in 2017. Thermal times for each crop variety are taken from a report published by Laine et al. (2017). Sowing dates and imaging dates are used to calculate the growth phase as a fraction of the total thermal time for the crop variety. Images with dates prior to 1st of July form the early data set and the remaining images the late one.

Field #	Size (ha)	Mean yield (kg/ha)	Crop (*Variety*)	Thermal time	Sowing date	Imaging date	Growth phase
1	5.96	5098	Wheat (*Zebra*)	1052	10 May	17 Aug	83%
2	10.26	6054	Barley (*Trekker*)	979.7	16 May	8 Jun	15%
						27 Jul	64%
3	2.97	8971	Barley (*Trekker*)	979.7	17 May	8 Jun	15%
						27 Jul	64%
4	13.05	4673	Barley (*RGT Planet*)	982.2	15 May	6 Jul	42%
5	4.66	6482	Barley (*Propino*)	981.4	15 May	15 Jun	22%
6	7.29	6884	Barley (*Propino*)	981.4	15 May	15 Jun	22%
7	10.92	7568	Barley (*Harbinger*)	976.3	24 May	6 Jul	36%
8	15.28	7585	Barley (*Trekker*)	979.7	18 May	1 Jun	10%
						13 Jul	49%
9	18.86	6991	Wheat (*KWS Solanus*)	1065	13 May	15 Jun	21%
						6 Jul	72%

Table 2
Number of data frames extracted from each field using frame sizes of 10 m, 20 m and 40 m. The number of frames decreases slightly with increasing frame size due to field edge effects.

Field #	10 × 10 m data frames	20 × 20 m data frames	40 × 40 m data frames	Mean data frame count
1	761	745	735	*747*
2	1102	1159	1150	*1137*
3	783	731	691	*735*
4	1494	1486	1454	*1478*
5	610	586	590	*595*
6	942	931	916	*930*
7	1240	1247	1224	*1237*
8	3736	3786	3812	*3778*
9	4556	4548	4520	*4541*
∑	**15224**	**15219**	**15092**	*15178*

with additional information on other key elements of CNNs such as batch normalization and regularization. We evaluated various setups of these CNN elements to find the best-performing algorithm and assess its performance in crop yield prediction.

2.2.1. Convolution operation

The convolution operation is the first of multiple transformations performed in a convolutional layer of CNNs. Generally, the convolution operation can be described as calculating the sum of products between a set of input values and values of a convolutional *kernel*, also called a *filter*. In CNN, the kernel values are trained to find optimal features from the point of view of the task to be solved (in our case, predicting crop yield). The operating principle of the kernel is depicted in Fig. 4 and the position of convolutional layers in the overall structure of the CNN used in this study can be seen from Fig. 7.

2.2.2. Batch normalization

While not a requirement for CNNs, the state-of-the-art is to apply batch normalization (Ioffe and Szegedy, 2015) as a constituent of deep learning model layers. Batch normalization is an optimization strategy

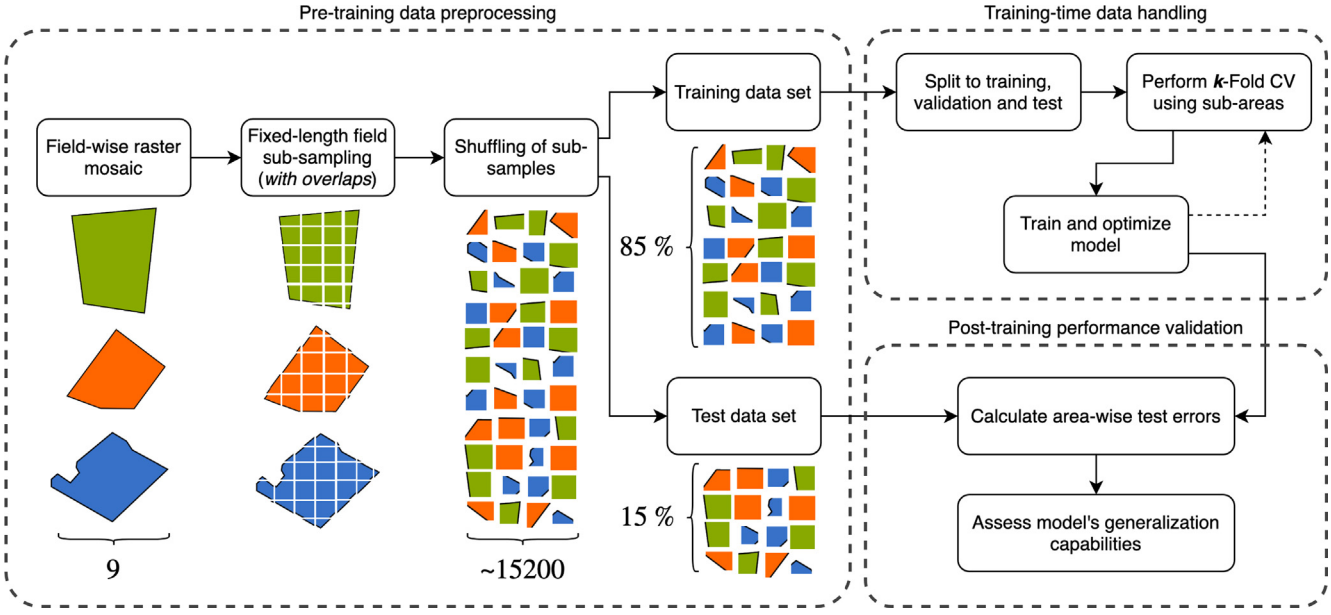

Fig. 1. All nine fields were first split to overlapping data frames of sizes 10 m, 20 m and 40 m. A dedicated holdout test data set was then built from 15% of shuffled data frames; these data were never presented to the model during training. The remaining 85% of data frames were then used for training the models with *k*-Fold Cross Validation. After the training phase of each model was completed, the test errors were calculated using the holdout test data set to validate the performance of the trained model.

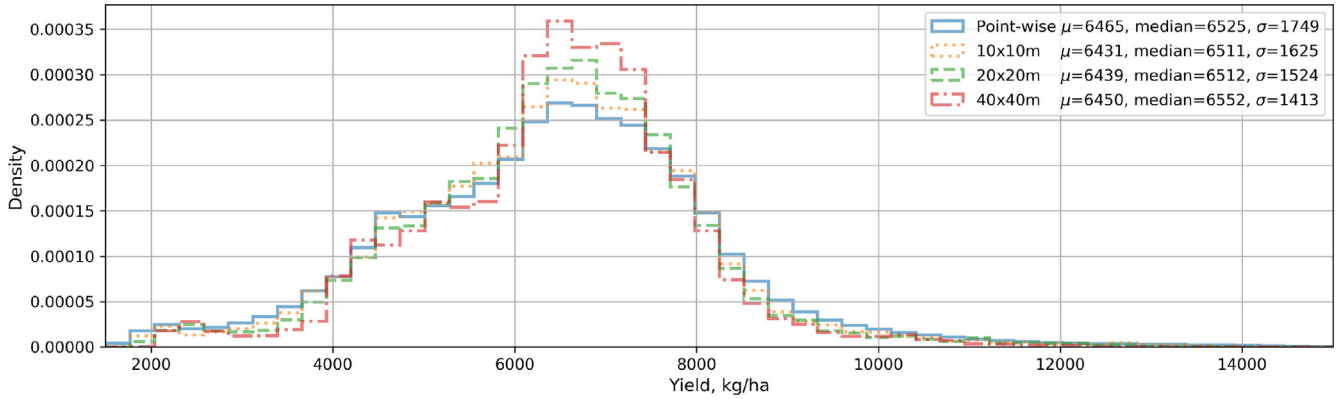

Fig. 2. Histograms and statistics of point-wise and window-averaged yield data. The histograms are normalized to probability densities to make point-wise graphs align with sliding window histograms count-wise. While sliding windows contain no-data points near field edges, only points containing data were taken into account.

for training deep models more efficiently. Batch refers to a subset of training data used for updating the model parameters (including kernel values) at a single iteration, albeit the term mini-batch is generally used to distinguish the whole data set (batch) from it's subset (mini-batch). It has been shown that normalizing the network layers for each batch (or mini-batch) of data stabilizes the learning, allowing to use higher learning rates and thus resulting in faster learning (Goodfellow et al., 2016). There are different implementations of batch normalization; the implementation used in the CNN of this study follows Eq. (1), where x is a mini-batch of activations, ϵ a non-significant constant to prevent numerical underflow, γ is the momentum and b is a layer-wise bias:

$$y = \frac{x - \mu_x}{\sigma_x + \epsilon} * \gamma + b. \quad (1)$$

2.2.3. Max pooling

The convolution operation is usually followed by pooling. Pooling means grouping of adjacent values using a selected aggregation function, which in our case was taking the maximum (hence max pooling) over the neighboring values within a predefined window. The step size of moving this window along the feature map is called *stride*. Pooling effectively diminishes the input image dimensions making the detected features more coarse and thus more robust to small variations (Goodfellow et al., 2016). The amount of dimension reduction is controlled by the stride parameter. The stride dictates how many applications of the pooling window are performed. An example of max pooling is given in Fig. 5 and the position of pooling in the overall structure of the CNN used in this study can be seen from Fig. 7.

2.2.4. Rectified linear units

A key element in any neural network is the layer-wise activation function of the neurons. A variety of activation functions have been designed, but the use of the rectified linear function in the activation units is the current standard for CNNs (He et al., 2015; Goodfellow et al., 2016). Activation units employing rectified linear functions are commonly referred to as ReLUs. The operating principle of this activation function is to allow only positive inputs to proceed linearly and is depicted in Fig. 6. We too use ReLUs as the activation functions in both the convolutional as well as the fully connected layers (see Fig. 7).

2.2.5. Fully connected layers

The convolutional layers of a CNN extract salient features from input images, i.e., factors with highest descriptive power regarding the data producing process. To utilize the learned features in a regression or a classification task, they have to be successfully mapped to a target value. This is performed typically by adding fully connected (FC) layers after the convolutional layers. The term *fully connected* refers to the principle that in these layers, each neuron (or unit) of the previous layer has a connection to each unit of the layer in question. Increasing the number of FC layers increases the capacity of the network to learn the mapping between the features and the target. It also increases the burden of optimization, as in FC layers the number of connections grows exponentially with the number of layers.

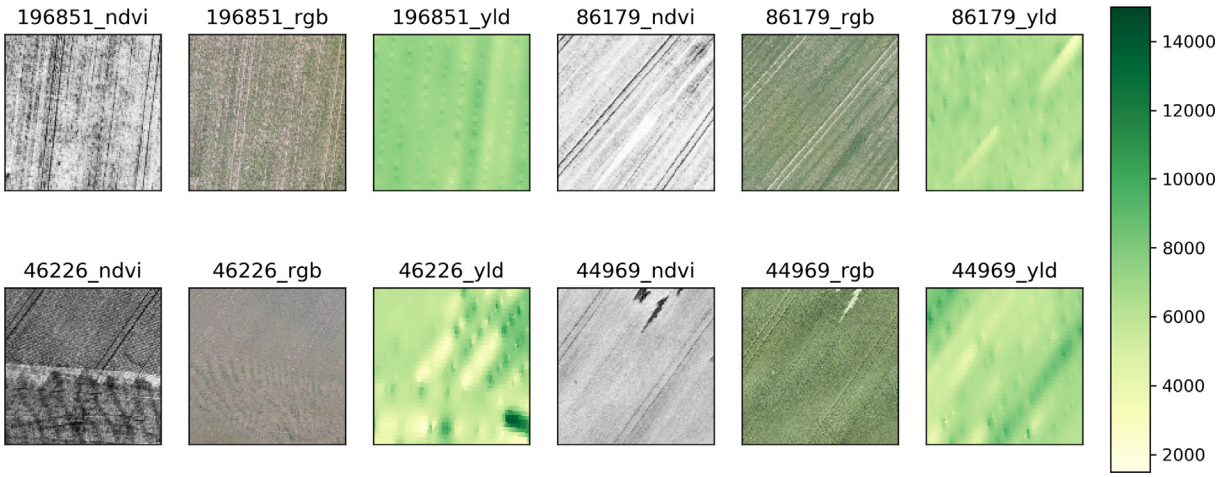

Fig. 3. Visualizations of NDVI and RGB input images and yield targets. The identification numbers above the images denote the distinct area from which the images were extracted.

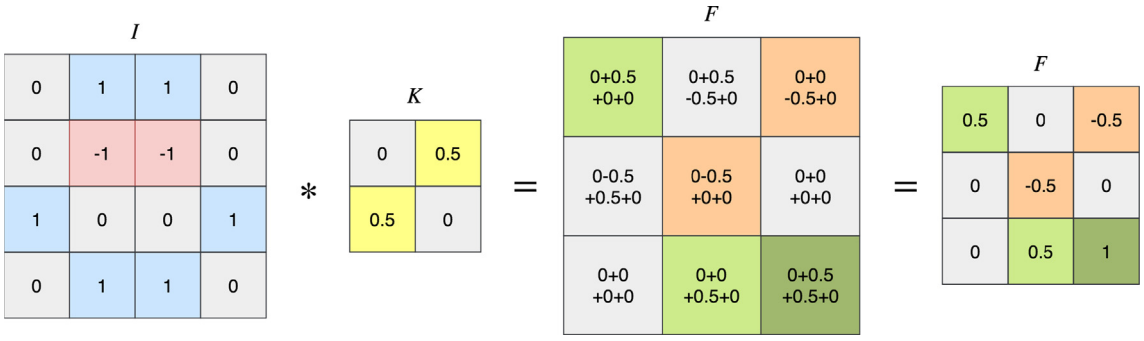

Fig. 4. The kernel K is applied to the input image I in a sliding window fashion. With each application, a sum of element-wise products is calculated and stored. After the kernel has been applied to the whole image, a complete feature map F is produced. A feature map indicates the result of detecting a kernel-specific feature in the input image.

2.2.6. Regularization strategies

Increasing the depth of a deep learning model allows it to learn more complex functions. This is also known as increased model's capacity (Goodfellow et al., 2016). When a model's capacity increases, it becomes more prone to overfitting to the training data in which case its ability to generalize (and, therefore, its performance on test data) deteriorates. This can be avoided with regularization, which effectively reduces the model's capacity diminishing the gap between training and test errors. Regularization is a comprehensive term for methods in machine learning that are used to lower the test error without focusing on training error.

In our model we make use of two distinct regularization strategies. First of the two is the L^2-penalty, also known as the weight decay. It diminishes the model's layer-wise parameters with each training iteration. When applied in conjunction with training by error back-propagation, the most relevant of the model's parameters retain their magnitude while non-relevant ones diminish. The second implemented regularization strategy is called early stopping. It is a robust meta-algorithm integrated into the training process to halt the training after n non-improving iterations. The hyperparameter n is called *patience* (Goodfellow et al., 2016).

2.2.7. Overall architecture

The basic architecture of the CNN implemented in this study follows closely the one reported by Krizhevsky et al. (2017). Their model performed extremely well in ImageNet Large Scale Visual Recognition Competition (Russakovsky et al., 2015) attaining top classification results in multiple categories. The general topology of our network is depicted in Fig. 7. The network was implemented using the PyTorch framework (Paszke et al., 2017). In our network we use non-overlapping pooling windows with pooling window size of 5 and a pooling stride matching the pooling window size. We also include the pooling function only in the first and the last convolutional layer. The reason for this is that at the lowest (i.e., in the case of 10 m ground resolution) our image size is 32 × 32 pixels and too many pooling operations would cause the data representation to collapse. This way our network is also scalable with respect to the number of layers. Regardless of the number

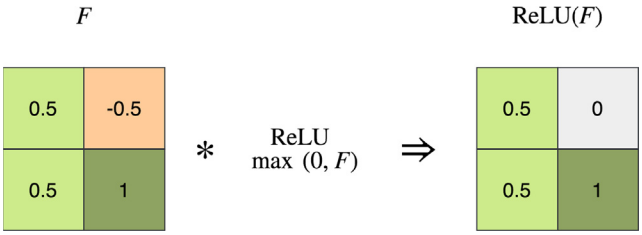

Fig. 6. An illustration of the effect of applying the rectified linear activation function to a pooled feature map.

of source image bands, our convolutional layers contain 64 kernels except for the last layer containing 128 kernels. Krizhevsky et al. (2017) incorporated two FC layers to the model with 2048 neurons per layer. We used similar number of layers with half the width, i.e., 1024 neurons per layer.

2.3. Optimizing the network

Finding the optimal configuration of any deep learning network is an iterative process, where the model's parameters are initialized and tuned multiple times. The goal is to find a set of model's parameters (weights, biases, etc.) and hyperparameters (learning rate, optimizer coefficients, etc.) that in conjunction produce the best performance. The output of the iterative process is a single model usually performing best when compared to other models produced within the process. We used absolute error between the network output and the target value (i.e., crop yield values) as the performance measure. In machine learning, the best performing model is considered to be the one that generalizes well to previously unseen data. To measure the generalization performance across training instances, we extracted and reserved a subset of data as a holdout test set. This test data set was used outside of the training loop to ensure that the model never learned from it. With the rest of the data we performed k-fold cross validation using three folds per epoch. An epoch is a single complete iteration over the full training data set consisting of windowed image samples of all 9

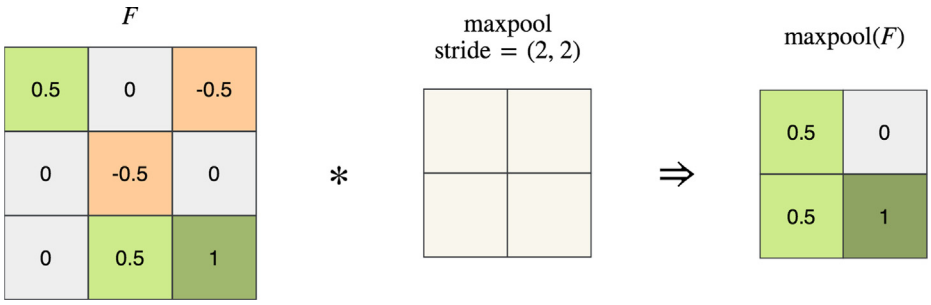

Fig. 5. An example of a simple application of max pooling, where the pooling is applied to a feature map F with pooling window size of 2 × 2 and a stride equaling the kernel size.

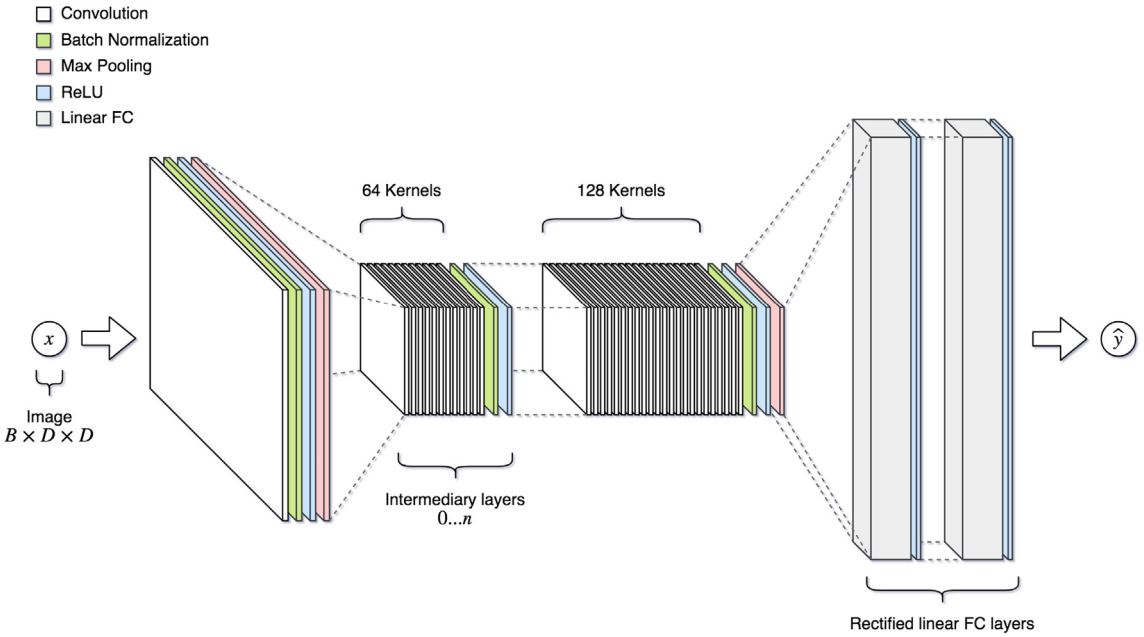

Fig. 7. The overall topology of the implemented CNN. Network's inputs can be single-band or multi-band images (B) with varying dimensions (D). The network has at least two convolutional layers accompanied with two fully connected layers. The depth of the network is controlled by the number of intermediary convolutional layers. The last convolutional layer has 128 kernels while the intermediary layers have 64 kernels. Max pooling is applied only in the first and last convolutional layers so that the size of the data representation stays consistent when network depth is varied.

fields.

The best training algorithm was evaluated among three options: Stochastic Gradient Descent with momentum (SGD-momentum) (Bottou, 1998), RMSprop (Hinton et al., 2014) and Adadelta (Zeiler, 2012). These training algorithms are suggested in Goodfellow et al. (2016) and they are also among the ones compared in Karpathy and Fei-Fei (2017). In a preliminary test, the three algorithms were tested for convergence by training the network for three epochs. Training was performed for each of the three data window sizes and each of the four sets of input data. The batch size was varied from 2^5 to 2^{10}. The worst performing algorithm was excluded and a second test performed on the remaining two by fixing the batch size to 128 (2^7) and training for 50 epochs, a number consistent across the training of almost every model.

The effect of the depth of the network on the performance was evaluated by training models with 4, 6, 8, 10 and 12 convolutional layers over 50 epochs per training session. The training was conducted for the NDVI and RGB images from early and late data sets and with all three input image dimensions using the previously selected training algorithm. At this stage, the best performing combination of - network depth, image type (NDVI or RGB) and window size - was selected based on error performance over the test data.

In the next step, the chosen training algorithm's hyperparameters (i.e., the learning rate and the past iterations' error correction adjustment) were tuned. In order to evaluate performance, benchmark models were created by initializing a model for each of the four data sets (i.e., early and late, RGB and NDVI). The hyperparameter values were searched over a coarse grid for values producing lowest test errors, followed by a more refined random search in the vicinity of the coarse minimum. Sensitivity of the network performance to initial values of the CNN parameters was also assessed.

In the last step, the hyperparameter combinations producing the best performance were used to test and tune the effect of regularization algorithms. Tuning of the weight decay coefficient (L^2 regularization) for early and late data sets was performed by searching over a coarse grid of values followed by refined search. Subsequently, the effect of early stopping was tested using values 10, 20, 30, 40 and 50 for the patience parameter (see Section 2.2.6).

3. Results

We measure the performance of the CNN by *mean absolute error*, i.e., the mean absolute difference between the true yield value and the CNN output (predicted value). This can also be called *loss*. We consider two different errors: the training error, obtained for the same data the network is trained with, and the test error, obtained for the data set aside for testing. The former one indicates how well the model is able to fit to the data, i.e., what is its capacity, while the latter one indicates how well the network is able to generalize to unseen data samples.

3.1. Selection of the training algorithm

Of the three training algorithms – Adadelta, SGD-momentum and RMSprop – the RMSprop showed poor convergence and was therefore ruled out from subsequent tests. Between the two remaining algorithms, Adadelta outperformed SGD-momentum and was chosen as the training algorithm for further experiments (see Table 3).

3.2. Depth of the network

In Fig. 8 the test and training errors for the three window sizes and for various networks depths are shown for the RGB data of earlier growth phase. The largest window (size 40 × 40 m) produced lowest test errors in majority of cases regardless of the network depth. The colored areas indicate gaps between training (lower bound of the area) and test (upper bound of the area) errors, also referred to as

Table 3
Lowest mean absolute test errors (kg/ha) observed among the three data window size configurations (10 m, 20 m and 40 m) with 50 epochs of training and a batch size of 128 samples for each source image type. Adadelta performed best with almost every source image configuration.

Optimizer	NDVI early	NDVI late	RGB early	RGB late
SGD with Momentum	1751.2	1183.7	1231.5	985.0
Adadelta	842.8	1165.1	836.2	989.5

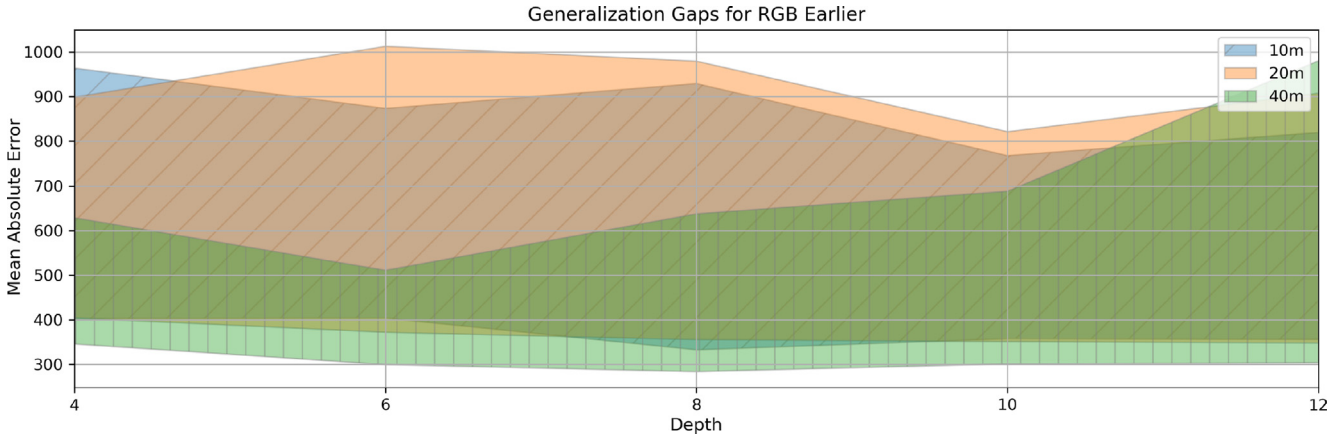

Fig. 8. The generalization gaps with early RGB images. The generalization gap is depicted as the difference between the training and the test errors. It shows how close the test error is to the effective capacity of the model, the training error. The lowest test errors (upper bound of the area) were achieved rather consistently with the source image window size of 40 × 40 m.

generalization gaps. The lowest test and training error combination is obtained with 6 convolutional layers. Also, the 40 m window and network depth of 6 convolutional layers result in the narrowest generalization gap.

3.3. Optimization of Adadelta hyperparameters

The hyperparameters of the chosen training algorithm, Adadelta, were tuned by considering the effects of the adaptive learning rate and the coefficient adjusting the effect of past iterations' error corrections (in the form of squared gradients) on learning. The latter is effectively similar to momentum, defining the magnitude by which the past affects the current learning process. The previous experiments were performed using default values for these hyperparameters, i.e., 1.0 for the learning rate, 0.9 for the coefficient for computing a running average of squared gradients, and 0 for the weight decay (see Table 4).

The initial grid search was conducted with hyperparameter values similar to those found in the original Adadelta research paper with an epoch limit of 50 compared to the original study's 6 epochs (Zeiler, 2012). For the early RGB data set, the optimal values were approximately 8×10^{-3} for the learning rate and 0.58 for the coefficient adjusting the effect of past iterations' error. For the late data set the respective values were 10^{-4} and 0.9. The effect of hyperparameter tuning on the performance of the network can be seen from the results in Table 4.

Table 4
The total improvement in test error compared to the benchmark model when using regularization and optimization of training algorithm hyperparameters. The benchmark models were trained with the early and late RGB image data with default parameters. Window size was 40 × 40 m. Errors are reported as mean absolute error (MAE) and mean absolute percentage error (MAPE). The best results are formatted in bold.

	RGB early		RGB late	
	MAE [kg/ha]	MAPE	MAE [kg/ha]	MAPE
Benchmark learning rate: 1.0 past err. coeff.: 0.9 weight decay: 0 patience: ∞	997.8	18.3%	1021.5	19.5%
with **Optimized Adadelta params.** learning rate (early/late): 0.008/0.0001 past err. coeff. (early/late): 0.58/0.9	546.2	9.6%	**624.3**	**11.4%**
and with **L^2-regularization** weight decay: 0.001	558.4	9.4%	700.4	13.1%
and with **Early Stopping** patience: 50	**484.3**	**8.8%**	680.4	12.6%

3.4. Optimization of the regularization parameters

The CNN models using optimal hyperparameters for the Adadelta training algorithm were trained next with early and late RGB data sets of 40 × 40 m window to determine the effect of regularization on the prediction error and to tune the regularization parameters. The tuning of weight decay coefficient with grid search first and zoomed-in random search after that resulted in the optimal coefficient value of 10^{-3} for both data sets. The optimal patience values were around 50, again for both data sets. It was observed that the increase in patience increased the training time significantly. The selected patience value allowed the models for both data sets to converge in approximately 250 epochs. The effect of using the L^2-regularization alone and combined with early stopping can be seen from Table 4.

4. Discussion and conclusions

This study presents a training paradigm of a CNN based deep learning model for predicting wheat and barley yield. The results indicate that the best performing model can predict within-field yield with a mean absolute error of 484 kg/ha (MAPE: 8.8%) based only on RGB images in the early stages of growth (< 25% total thermal time). The model for RGB images at later growth stage returned higher error values (MAE: 680 kg/ha; MAPE: 12.6%). In searching for optimal performance, the input data window size (10 m, 20 m, 40 m), the data acquisition time (early vs. late) and data modality (RGB vs. NDVI) were varied. The 9 fields included in the study were imaged by a camera mounted to UAV and together taken as a source of > 10,000 input image frames covering a total area of 90 hectares. Network depth (i.e., the number of convolutional layers), the training algorithm and its hyperparameters as well as the CNN regularization scheme were also optimized. The lowest error was achieved using a network consisting 6 convolutional layers followed by two fully connected layers regularized with L^2-regularization coefficient of 10^{-3} and early stopping patience of 50. The optimizer was also tuned for the optimal value of the learning rate (8×10^{-3}) and the coefficient adjusting the effect of past iterations' error corrections (0.58). The results show that the lowest test errors were achieved with the largest data window size tested (40 m).

The training of any neural network is always influenced by the

combined randomness resulting from how the data is shuffled between cross validation folds, the optimization process and other factors. This in turn means that, while discrete error metrics produce a ranking across hyperparameter setups, slight variations between test errors can be attributed to the random nature of the optimization process as a whole. We optimized distinct models for early and late RGB data sets. The best performing model used RGB images from the early growing season and benefited from regularization. The model using the late RGB images didn't gain from added regularization, as the best performance was achieved during the tuning of the training algorithm (see Table 4).

In yield prediction, the shift from using traditional regression methods (Ruß, 2009) towards artificial neural network based methods (Chlingaryan et al., 2018) has resulted in improved performance (Jiang et al., 2004; Kaul et al., 2005). Among these ANN based studies, those using remote sensing image data to train their prediction models have achieved low prediction errors (≈ 5%). These models are specific to the crop types whose images they are trained with (e.g., soybean, wheat, rice). Jiang et al. (2004) working with satellite images reported an average relative winter wheat prediction error of 3.5%. The indices used for training the model were: NDVI, surface temperature, absorbed photosynthesis active radiation, water stress index and 10-year average crop yield. Bose et al. (2016) employed spiking neural networks to estimate winter wheat yield from satellite based NDVI images at the region level, achieving a best average relative error of 4.35%. In their recent work, You et al. (2017) leveraged advanced hybrid machine learning algorithms to achieve very low soybean yield prediction errors (3.19–5.65%) using only satellite images.

A commonality among these studies is the use of satellite imagery and large spatial scales of their analyses (region or county level predictions). Our study, in contrast, seeks to perform predictions at the intra-field scale using UAV based images in order to spatially analyze yield within the field. In one of the earliest studies on this topic, Davis and Wilkinson (2006) used satellite imagery of wheat crop (visible, infrared and radar) and an ANN model showing promising results (error slightly above 10%) for a single field (≈ 36 ha). Khanal et al. (2018) employed various machine learning algorithms (including neural networks) and aircraft based multi-spectral images to predict corn yield on a single field (17.5 ha). A few studies have applied ANN's for classifying crops (Rebetez et al., 2016) and yield (Pantazi et al., 2016) at the intra-field scale. However, rather than classifying within yield categories this study aims at quantitative predictions. Models at intra-field scale would offer the individual farmer the possibility of in-season monitoring of crop, which would enable decision support systems for interventions necessary to achieve higher yields. Models trained at large regional scales rarely extrapolate to finer scales, though efforts are underway to develop scalable models (Donohue et al., 2018). The methodology introduced by You et al. (2017) shows great potential and as authors claim its scalability, it would certainly be of interest in testing at the intra-field scale.

One important aspect of remote sensing based yield prediction has been finding image channels or indices containing the most discriminating features necessary for analysis (Panda et al., 2010). Consequently, the finding in this study that the RGB images perform better than NDVI, assumes significance and aligns with the study for estimating biomass and crop height (Näsi et al., 2017). This indicates that multiple spectral bands increase the information content in comparison to the condensed NDVI image. From a utility perspective, RGB cameras are cheaper with most commercially available UAVs already fitted with decent cameras able to produce images of high resolution. Models that can perform well without the need for expensive specialized equipment will make the analyses accessible to an individual farmer.

The relationship between crop yield and its environment is non-linear and may not be sufficiently contained in the features captured by images. As shown by the studies reporting low prediction error levels, by adding multi/hyper spectral data, temporal image data, soil and environmental features in the feature matrix, it is possible to constrain the resulting model error effectively. Considering that this study models the yield based only on images, the resulting prediction error of 8.8% is promising. Additionally, collection of multi-year yield maps from sensor-equipped harvesters would add valuable information to act as ground truth. More than 90 hectares of fields were mapped in this study (2017 season). In 2018 a similar set of data has been acquired while the data acquisition will be continued in 2019. This valuable database will serve to further train, tune and verify the current model for greater accuracy. An additional limitation of this study is that only minimal preprocessing was applied to the source data. Developing automated error correction methods for data preprocessing would be another important task when developing remote sensing based crop yield models. Careful artifact rejection and preprocessing would probably benefit the modeling considerably.

In conclusion, this study is an important step towards establishing a combined model for wheat and barley yield prediction in the Finnish continental subarctic climate. The long summer growing days in this region presents a unique profile of temperature and photoperiod, justifying a region specific deep learning model for these crops. By collecting data using commercial off-the-shelf UAV and camera packages, we focus our attention on a spatial scale that enables us to predict intra-field yield distribution within the context of individual farm crop monitoring. The results indicate that the CNN models are capable of reasonable accurate yield estimates based on RGB images. It is worth noting that the CNN architecture seemed to be performing better with RGB images than NDVI images. In the future, the developed model will be trained on a larger set of features (climate and soil) along with time series image data to tune the trained model for accuracy.

Acknowledgments

We would like to give special acknowledgments to Mtech Digital Solutions Oy for partly funding this research. We also want to thank the MIKÄ DATA project's research group of Tampere University of Technology for providing the data and additional knowledge required to use the data appropriately. A special thanks to Mikko Hakojärvi from Mtech Digital Solutions Oy for providing insight into the agricultural knowledge domain.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compag.2019.104859.

References

Bose, P., Kasabov, N.K., Bruzzone, L., Hartono, R.N., 2016. Spiking neural networks for crop yield estimation based on spatiotemporal analysis of image time series. IEEE Trans. Geosci. Remote Sens. 54 (11), 6563–6573.

Bottou, L., 1998. On-line Learning in Neural Networks. Cambridge University Press, New York, NY, USA Ch. On-line Le, pp. 9–42.

Chlingaryan, A., Sukkarieh, S., Whelan, B., 2018. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: a review. Comput. Electron. Agric. 151 (November 2017), 61–69.

Chunjing, Y., Yueyao, Z., Yaxuan, Z., Liu, H., 2017. Application of convolutional neural network in classification of high resolution agricultural remote sensing images. ISPRS – Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-2/W7, 989–992.

Davis, I.C., Wilkinson, G.G., 2006. Crop yield prediction using multipolarization radar and multitemporal visible/infrared imagery. In: Proc.SPIE 6359, 6359–6359 – 12.

Donohue, R.J., Lawes, R.A., Mata, G., Gobbett, D., Ouzman, J., 2018. Towards a national, remote-sensing-based model for predicting field-scale crop yield. Field Crops Res. 227, 79–90.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision 2015 Inter, pp. 1026–1034.

Hinton, G., Srivastava, N., Swersky, K., 2014. Neural Networks for Machine Learning Lecture 6a: Overview of minibatch gradient descent.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

Jiang, D., Yang, X., Clinton, N., Wang, N., 2004. An artificial neural network model for

estimating crop yields using remotely sensed information. Int. J. Remote Sens. 25 (9), 1723–1732.
Kamilaris, A., Kartakoullis, A., Prenafeta-Boldú, F.X., 2017. A review on the practice of big data analysis in agriculture. Comput. Electron. Agric. 143, 23–37.
Karpathy, A., Fei-Fei, L., 2017. Deep visual-semantic alignments for generating image descriptions. IEEE Trans. Pattern Anal. Mach. Intell. 39 (4), 664–676.
Kaul, M., Hill, R.L., Walthall, C., 2005. Artificial neural networks for corn and soybean yield prediction. Agric. Syst. 85 (1), 1–18.
Khanal, S., Fulton, J., Klopfenstein, A., Douridas, N., Shearer, S., 2018. Integration of high resolution remotely sensed data and machine learning techniques for spatial prediction of soil properties and corn yield. Comput. Electron. Agric. 153 (August), 213–225.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90.
Laine, A., Högnäsbacka, M., Niskanen, M., Ohralahti, K., Jauhiainen, L., Kaseva, J., Nikander, H., 2017. Virallisten lajikekokeiden tulokset 2009-2016, 262.
Matikainen, L., Karila, K., Hyyppä, J., Puttonen, E., Litkey, P., Ahokas, E., 2017. Feasibility of multispectral airborne laser scanning for land cover classification, road mapping and map updating. ISPRS - Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-3/W3, 119–122.
Milioto, A., Lottes, P., Stachniss, C., 2017. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. Adv. Intell. Syst. Comput. 531, 105–121.
Miyoshi, G.T., Imai, N.N., de Moraes, M.V.A., Tommaselli, A.M.G., Näsi, R., 2017. Time series of images to improve tree species classification. ISPRS – Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-3/W3, 123–128.
Näsi, R., Viljanen, N., Kaivosoja, J., Hakala, T., Pandžić, M., Markelin, L., Honkavaara, E., 2017. Assessment of various remote sensing technologies in biomass and nitrogen content estimation using an agricultural test field. ISPRS – Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-3/W3, 137–141.
Panda, S.S., Ames, D.P., Panigrahi, S., 2010. Application of vegetation indices for agricultural crop yield prediction using neural network techniques. Remote Sens. 2 (3), 673–696.
Pantazi, X.E., Moshou, D., Alexandridis, T., Whetton, R.L., Mouazen, A.M., 2016. Wheat yield prediction using machine learning and advanced sensing techniques. Comput. Electron. Agric. 121, 57–65.
Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in PyTorch. In: NIPS-W.
Rebetez, J., Satizábal, H.F., Mota, M., Noll, D., Büchi, L., Wendling, M., Cannelle, B., Pérez-Uribe, A., Burgos, S., 2016. Augmenting a convolutional neural network with local histograms - a case study in crop classification from high-resolution uav imagery. In: 24th European Symposium on Artificial Neural Networks, ESANN 2016, Bruges, Belgium, April 27–29, 2016.
Ruß, G., 2009. Data mining of agricultural yield data: a comparison of regression models. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5633 LNAI, 24–37.
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vision 115 (3), 211–252.
Sa, I., Chen, Z., Popovic, M., Khanna, R., Liebisch, F., Nieto, J., Siegwart, R., 2017. weedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming.
Tiusanen, J., 2017. Aineiston käsittely ja muotoilu. Käytännön Maamies.
Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M.J., 2017. Big data in smart farming – a review. Agric. Syst. 153, 69–80.
You, J., Li, X., Low, M., Lobell, D., Ermon, S., 2017. Deep Gaussian process for crop yield prediction based on remote sensing data. In: 31th AAAI Conference on Artificial Intelligence, pp. 4559–4565.
Zeiler, M.D., 2012. ADADELTA: An Adaptive Learning Rate Method. undefined.