Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Plant species classification using deep convolutional neural network

Mads Dyrmann [a,*], *Henrik Karstoft* [b], *Henrik Skov Midtiby* [a]

[a] *The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Denmark*
[b] *Department of Engineering, Aarhus University, Denmark*

ARTICLE INFO

Article history:
Received 16 March 2016
Received in revised form
22 August 2016
Accepted 26 August 2016
Published online 13 September 2016

Keywords:
Plant classification
Deep learning
Convolutional Neural Networks
Weed control

Information on which weed species are present within agricultural fields is important for site specific weed management. This paper presents a method that is capable of recognising plant species in colour images by using a convolutional neural network. The network is built from scratch trained and tested on a total of 10,413 images containing 22 weed and crop species at early growth stages. These images originate from six different data sets, which have variations with respect to lighting, resolution, and soil type. This includes images taken under controlled conditions with regard to camera stabilisation and illumination, and images shot with hand-held mobile phones in fields with changing lighting conditions and different soil types. For these 22 species, the network is able to achieve a classification accuracy of 86.2%.

1. Introduction

In modern farming, there is a need for effective weed control as yield losses caused by not controlling the weeds are significant (23%–71% depending on the crops, weeds and location (Hodgson, 1968; Oerke, 2006; Slaughter, Giles, Fennimore, & Smith, 2008)). The preferred method used for controlling weeds is to apply herbicides to the field, as it is cheap and effective compared to mechanical weeding. There is, however, a growing governmental pressure on farming to limit the usage of herbicides because of environmental concerns. Many new approaches for weed control, which reduces the herbicide consumption, needs additional information about the weed plant population.

Weed control methods, which rely on mechanical means, need to know the precise location of crop plants, while methods that optimise the herbicide application requires information on the species of the weed plants. By using detailed information about present weed plant species, their growth stages and plant densities in a field, the herbicide consumption can be reduced by 40% on average (Jørgensen et al., 2007). The Danish system, *Crop Protection Online,* is a tool that helps farmers choose optimal herbicide mixtures given a certain weed infestation. This system operates with 104 weed species.

Image processing has previously been used to solve the task of recognising different weeds and crops. Noticeable studies trying to solve this task are listed in Table 1. These studies include Søgaard (2005), who was able to classify Shepherd's

[*] *Corresponding author.*
E-mail address: mady@mmmi.sdu.dk (M. Dyrmann).

http://dx.doi.org/10.1016/j.biosystemseng.2016.08.024
1537-5110/© 2016 IAgrE. Published by Elsevier Ltd. All rights reserved.

Nomenclature

β	trainable scale for network layer
γ	trainable bias for network layer
μ	mean of image batch
σ	standard deviation of image batch
x	image batch
BBCH	Biologische Bundesanstalt, Bundessortenamt und CHemische Industrie
MSRA	Microsoft Research Lab - Asia
ReLU	The rectified linear unit
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
TLR	Twin Leaf Region
VGG	Oxford Visual Geometry Group

purse (*Capsella bursa-pastoris*), Scentless mayweed (*Tripleurospermum inodorum*) and Charlock (*Sinapis arvensis*) with an accuracy of between 65% and 93% by using active shape models.

Åstrand and Baerveldt (2002) classified sugar beet in fields. They distinguished sugar beet from weeds by using only colour features and achieved an accuracy of 91%.

Plants can also be recognised by using shape based features. Giselsson (2010) used shape based features to classify 1700 plant samples of eight species, where each species was represented solely at one growth stage. From an overall classification accuracy of 94.8% was achieved.

Dyrmann and Christiansen (2014) demonstrated a framework that was able to distinguish seven weed and crop species at different early growth stages by fusing a shape based classification of leaves and whole plants. They achieved an overall accuracy of 95.8%.

Kazmi (2014) used local features for plant recognitions rather than using features, that describe the whole shape of plants. These local features include Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) and Twin Leaf Region (TLR) features. By using SIFT features to describe interest points found using Hessian-Laplace, he was able to distinguish thistles and sugar beet with an accuracy of 99%.

Golzarian and Fricka (2011) extracted different colour and shape features from wheat, ryegrass and brome grass at early growth stages and used principal component analysis to discriminate these species. A classification accuracy of between 82.4% and 88.2% was achieved.

Table 1 – Some papers on plant recognition, the number of distinguished species and classification accuracy.

Source	#Species	% Correct	Approach
Søgaard (2005)	3	65%–93%	Active shape models
Åstrand and Baerveldt (2002)	2	91%	Colour features
Giselsson (2010)	8	94.8%	Shape and colour features
Dyrmann and Christiansen (2014)	7	95.8%	Shape and colour features
Kazmi (2014)	2	99%	Local features
Golzarian and Fricka (2011)	3	82.4%–88.2%	Shape and colour features

One of the obstacles of these studies is the limited number of plant species that can be distinguished; this might be enough for locating crop plants for mechanical weeding, but it is insufficient for choosing optimal herbicide mixtures. Thus there is a need for new techniques which can handle a larger number of plant species.

This paper demonstrates one approach for designing and training a deep convolutional neural networks to distinguish between a large number of plant species. The main idea of a convolutional neural network is to build a hierarchy of self-learned features, all of which are based on less abstract features from previous layers of the network. Compared to precious classification methods, these self-learned features make the convolutional neural network less affected by natural variations such as changes in illumination, shadows, skewed leaves and occluded plants. Furthermore, segmentations of plants from the soil is not a necessary preprocessing step for the classification method. Because the deep convolutional neural network is able to find image features by itself; the network is able to learn new plant species with little effort since the need for designing new feature descriptors is removed.

Convolutional neural networks have received much attention in recent years as they have proven capable of outperforming previous records in image recognition challenges. Most noticeably is the work by Krizhevsky, Sutskever, and Hinton (2012), who in 2012 set the record in *ImageNet Large Scale Visual Recognition Challenge* (Russakovsky et al., 2015) with a margin of 10.9% compared to the second-best entry. Whereas the *ImageNet* challenge is about separating objects from 1000 different categories (e.g. cat, car, tree, house …), the aim of this study is to perform a fine-grained separation of seedlings in their respective species. Compared with *ImageNet*, there is not much variation in the circumstances of how images in this study are taken. However, this is counterbalanced by the fact that there is also not much variation in the visual appearance between the different classes.

2. Data material

The network was trained and tested on images that were photographed vertically towards the ground. The images contained 22 different plants species at early growth stages. Six image datasets were joined, all covering only the early growth stages of the plants, mainly from BBCH 12–16 (Meier, 2001). The six datasets contained both images acquired under controlled lightning and images collected with cell phones in the field under changing lightning conditions. The datasets are listed in Table 2.

The plants were between 2 and 10 days old and typically between 10 mm and 40 mm from one leaf-tip to another. Some plants were grouped into families rather than individual species. The reason for this grouping is that these species have the same appearance and therefore are difficult to label at this growth stage. Examples of this are grasses that were grouped into narrow-leaved and broad-leaved grasses and plants in the Polygonum family, that were also grouped together.

Table 2 – The six datasets, listed with the number of species and the number of samples.

#		Species	Samples
1.	Dyrmann and Christiansen (2014)	12	5.539
2.	Robo Weed Support (2015)	13	1.630
3.	From Kim Andersen and Henrik Midtiby	7	1.447
4.	Søgaard (2005)	16	745
5.	Scharr, Minervini, Fischbach, and Tsaftaris (2014); Minervini, Abdelsamea, and Tsaftaris (2013)	2	284
6.	Aarhus University - Department of Agroecology and SEGES (2015)	17	62

Table 3 shows the 22 classes and the number of samples for each class. Samples from each of the different classes can be seen in Fig. 1.

The images were divided in a training and a test set with approximately 60% of the images put into the training set.

3. Methods

This section describes the preprocessing of the images, as well as the architecture of the convolutional neural network that was used for classification of the plants.

3.1. *Data augmentation*

The deep neural network was only given the raw pixel values rather than feature descriptors, which means that, unlike most traditional shape-based classification systems, this network had no prior knowledge of what is plant and what is background. This could potentially lead to problems as not all species were present in all datasets. The network could therefore learn to recognise the background, which will cause the network to be biased towards some classes When it detects specific backgrounds, which obviously is undesirable. Therefore, a simple excessive green segmentation (Woebbecke, Meyer, Bargen, & Mortensen, 1995) was used to detect green pixels. All non-green pixels were then removed before the images were fed into the network. This means that the backgrounds were the same for all images and thus they could not be used for the determination of plant species.

Table 3 – List of species and the number of samples before they are split in training and test sets. The last column indicates in which of the six datasets, the samples are found.

#	Name (*Latin*)	No. of samples	Datasets
0	Sherpherd's-Purse (*Capsella bursa-pastoris*)	420	1,2,4,6
1	Chamomile (*Matricaria*)	1378	1,2,4,6
2	knotweed family (*Polygonaceae*)	363	2,4,6
3	Cranesbill (*Geranium spp.*)	796	1,2,4,6
4	Chickweed (*Stellaria media*)	845	1,2,4,6
5	Veronica (*Veronica*)	132	2,4,6
6	Fat-Hen (*Chenopodium album*)	787	1,2,3,4,6
7	Narrow-leaved grasses (*Poaceae*)	1514	1,2,4,6
8	Field Pancy (*Viola arvensis*)	123	2,4,6
9	Broad-leaved grasses (*Poaceae*)	154	2,6
10	Annual Nettle (*Urtica urens*)	90	2,4,6
11	Black Nightshade (*Solanum nigrum*)	308	2,3,4,6
12	Cabbage family (*Brassicaceae*)	1171	1,2,3,4,6
13	Tobacco (*Nicotiana*)	83	5
14	Thale Cress (*Arabidopsis thaliana*)	201	5
15	Cleavers (*Galium aparine*)	355	1,4,6
16	Common Poppy (*Papaver rhoeas*)	79	4,6
17	Cornflower (*Centaurea cyanus*)	336	3,4,6
18	Wheat (*Triticum aestivum*)	253	1
19	Maize (*Zea Maize*)	293	1,3
20	Sugar Beet (*Beta vulgaris*)	463	1
21	Barley (*Hordeum vulgare*)	269	3
Total		10,413	

Because of this segmentation, some of the plants were split in multiple parts. This included some plants in the knotweed family (*Polygonaceae*), where the segmentation method removes their red stems. The network was therefore likely to find features, that are valid for the segmented images, but not unsegmented images. However, segmentation should be omitted if this method is to be applied to a unmonitored application where vast amount of training images is available. This is because the segmentation is sensitive to lightning and algae on the soil and therefore is expected to lower the overall performance.

A convolutional neural network is translation invariant but not rotation invariant. The plants, however, are photographed vertically towards the ground and therefore had no fixed orientation. We can therefore generate more training data by rotating the original training data. As plants were assumed to be symmetric, even more training data can be generated by mirroring the training data. The training set was thereby increased eight-fold by mirroring the images and rotating them in 90° increments. After the augmentation of the training data, there were 50,864 training samples.

The network required all training images to have the same size. Therefore, after the segmentation, padding was added to make all images square and the images were then scaled to 128 × 128 pixels. This scaling of the images removed the correlation between image size and physical size of the plants, which meant that images from the different datasets could be mixed despite being acquired with different cameras at different heights.

3.2. *Model architecture*

Several pre-trained networks for image classification exist, but they were usually trained on images that are very different from the images in this case. Most of the features learned by these networks were therefore not useful for distinguishing plants where they are expected to respond to a very limited amount of the features learned by the pre-trained models. Instead, a new architecture was built, which incorporated some of the newest findings in the domain.

The building blocks of the network were:

3.3. *Convolutional layers*

Convolutions are the main building block of a convolutional neural network. Filter kernels are slid over the image and for

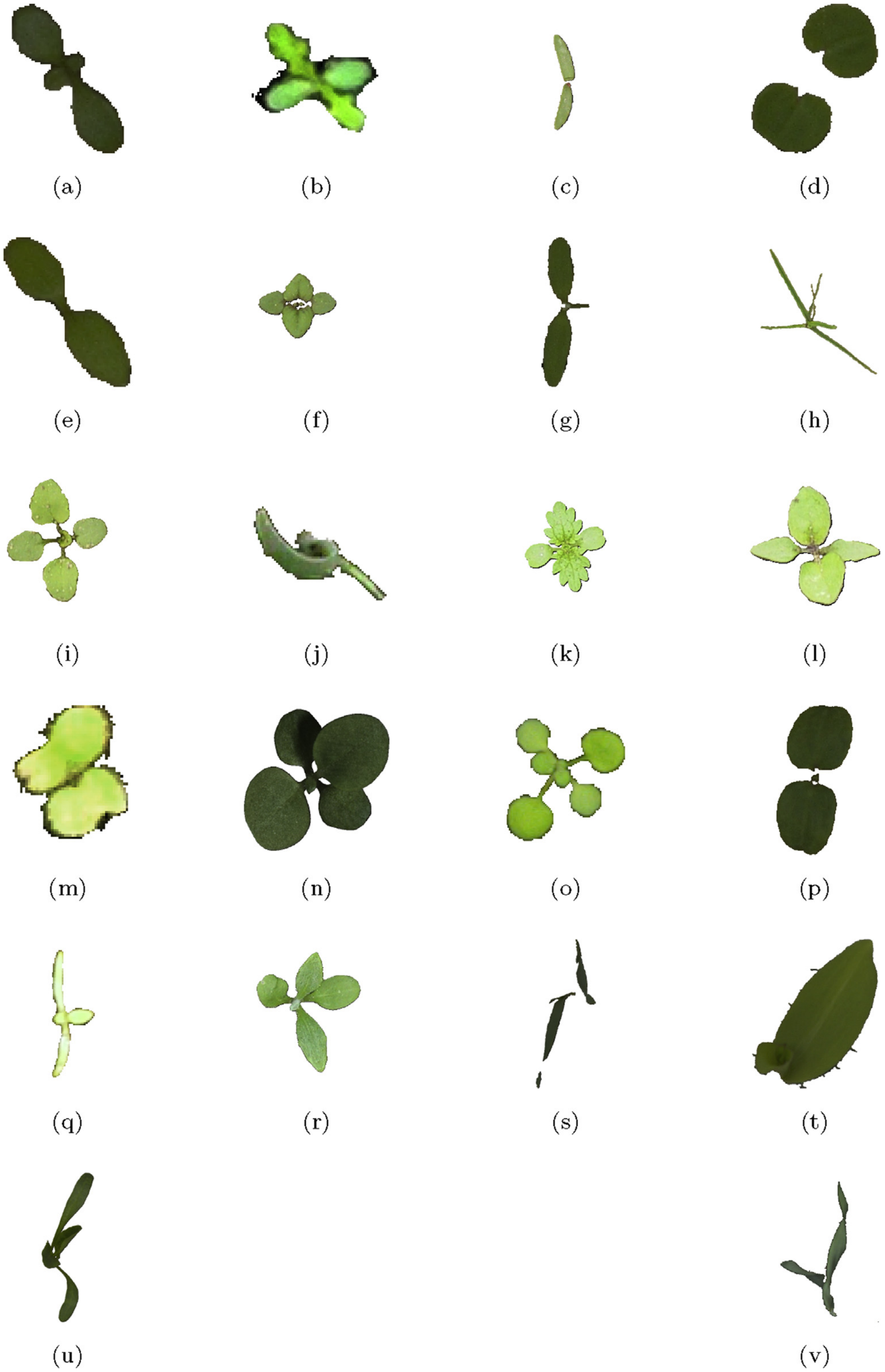


Fig. 1 – Segmented samples of the 22 classes. (a) Sherpherds-Purse, (b) Chamomile, (c) knotweed, (d) Cranesbill, (e) Chickweed, (f) Veronica, (g) Fat-Hen, (h) Narrow-leaved grasses, (i) Field pancy, (j) Broad-leaved grasses, (k) Annual nettle, (l) Black nightshade, (m) Cabbage family, (n) Tobacco, (o) Thale cress, (p) Cleavers, (q) Common poppy, (r) Cornflower, (s), Wheat, (t) Maize, (u) Sugar beet, (v) Barley.

each position the dot product between the filter kernel and the part of the image covered by the kernel is taken.

3.4. *Batch normalisation*

Batch normalisation ensures that the inputs to layers always fall in the same range even thought the earlier layers are updated. According to Ioffe and Szegedy (2015), batch normalisation results in a significant reduction in the required number of training iterations, but the same results are obtained as the network without normalisation. Batch normalisation is applied by normalising the output of a given layer to its standard deviation as shown in Eq. (1) (Ioffe & Szegedy, 2015).

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \in}} \gamma + \beta \tag{1}$$

where μ and σ are the mean and standard deviation of the current batch x, and γ and β are trainable parameters, that are updated slightly after each batch, ε is a small constant that is added to the variance to avoid zero-division. During tests, the mean, μ, and standard deviation, σ, are not calculated based on the test data, as it might give problems if the test set is small. Instead, μ and σ are set to the average statistics for the training data.

3.5. *Activations functions*

An activation function introduces non-linear decision boundaries to the network.

The rectified linear unit (ReLU) is an often used activation function in deep learning applications, as it is considerably faster to calculate than alternatives such as the sigmoid function, while still providing good results. The ReLU function is defined as:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

3.6. *Max-pooling layers*

Max-pooling is a process, which reduces the spatial size of a feature map and provides translation invariance to the network. This is done by only keeping the maximum value within a $k \times k$ neighbourhood in the feature map.

3.7. *Fully connected layers*

In a fully connected layer all inputs are connected to all outputs of the previous layer. This is known from traditional neural networks. Thus, a fully connected layer causes the spatial information to be removed. In convolutional neural networks, fully connected layers are usually used as a way of mapping spatial features to image labels.

3.8. *Residual layers*

A residual layer is a concept introduced in the MSRA architecture (He, Zhang, Ren, & Sun, 2015), that uses "shortcuts" to help a network converge in spite of the depth. These shortcuts work as identity mappings that bring low-level images features to the higher abstraction layers. According to He et al. (2015, pp. 171–180), the shortcuts help the network propagate to a lower error rate than the "plain" counterpart of the network.

In this study, the number of layers for the network was determined by evaluating the filter capacity and coverage of the network (Cao, 2015). Our convolutional neural network is sketched in Fig. 2.

3.8.1. *Filter capacity*

The filter capacity is a measure of how well a filter is able to detect complex structures in images. If the capacity is small, it means that only local features in the image are mapped to the next layer. On the other hand, if the capacity is large, it means that the filter is able to find complex structures of elements that are not neighbours in the input image. For example, in the case of small plants, there will often be two leaves opposite each other. This could be detected by a high-capacity filter, but not a filter with a low capacity as illustrated in Fig. 3.

The filter capacity is calculated as the ratio between the *real filter size* and the *receptive field* (Cao, 2015, pp. 1–6).

$$\mathit{Capacity} = \frac{\text{real filter size}}{\text{receptive field}} \tag{3}$$

where the *real filter size* is the size of the kernel in which *downsampling* (i.e. striding or pooling) of previous layers is taken into account. If no *downsampling* is applied, the *real filter size* is the same as the kernel size. For example if the input to a layer with kernel size $n \times n$ is *downsampled* by a factor k, the *real filter size* would then be $kn \times kn$. In this network there are three 2 × 2 max-pooling layers. After the first max-pooling layer, the *real filter size* would be $2n \times 2n$. After the second max-pooling layer it would be $2^2n \times 2^2n$ and after the third max-pooling layer it would be $2^3n \times 2^3n$. The *receptive field* is the portion of the original image, that the filter can "see", when following one path back through the network. The receptive field and thus the filter capacity can be increased by either increasing the size of filters in the convolutional layers or by using pooling.

Tests conducted by Cao (2015, pp. 1–6) showed that the capacity should not fall below 1/6. Because of the two shortcuts in the network, the capacity depends of which path through the network that is used. Since there are no operations on the two shortcuts, they will not contribute to an increased capacity. The capacity will therefore be calculated for the main branch, which ensures that the capacity is calculated from the maximum available receptive field. For this network, the filter capacity is between 37.7% and 100% and thereby well above the lower 1/6 limit.

3.8.2. *Coverage*

The coverage for a layer in a convolutional neural network is a measure of how big a part of the input image the layer can "see". Coverage can be increased by adding convolutional or pooling layers. At the end of the network, the coverage should not exceed 100%. If coverage exceeds 100%, it means that the network can handle images larger than the input image, which is a waste of calculations.

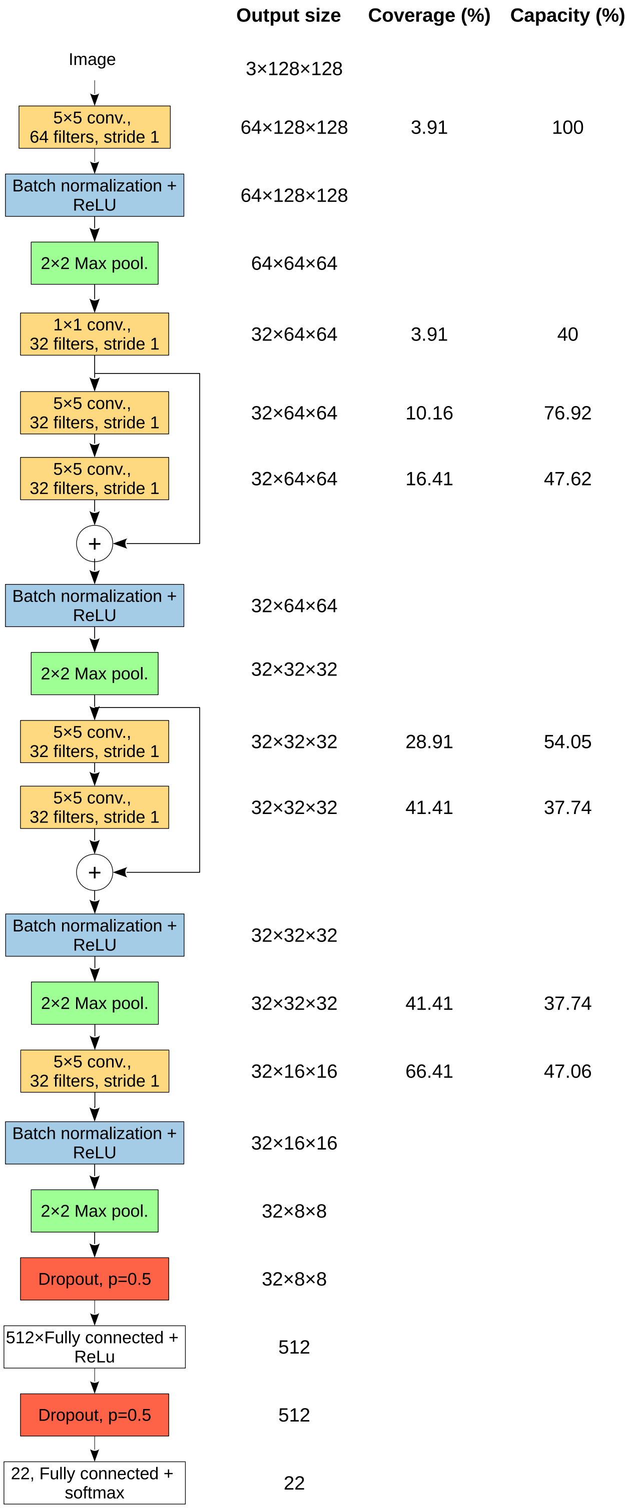

Output size
Coverage (%)
Capacity (%)
Image
3×128×128
5×5 conv., 64 filters, stride 1
64×128×128
3.91
100
Batch normalization + ReLU
64×128×128
2×2 Max pool.
64×64×64
1×1 conv., 32 filters, stride 1
32×64×64
3.91
40
5×5 conv., 32 filters, stride 1
32×64×64
10.16
76.92
5×5 conv., 32 filters, stride 1
32×64×64
16.41
47.62
+
Batch normalization + ReLU
32×64×64
2×2 Max pool.
32×32×32
5×5 conv., 32 filters, stride 1
32×32×32
28.91
54.05
5×5 conv., 32 filters, stride 1
32×32×32
41.41
37.74
+
Batch normalization + ReLU
32×32×32
2×2 Max pool.
32×32×32
41.41
37.74
5×5 conv., 32 filters, stride 1
32×16×16
66.41
47.06
Batch normalization + ReLU
32×16×16
2×2 Max pool.
32×8×8
Dropout, p=0.5
32×8×8
512×Fully connected + ReLu
512
Dropout, p=0.5
512
22, Fully connected + softmax
22

Fig. 2 – Network architecture. Two shortcuts are made: One from input of third layer to output of fourth layer, and one from input to fifth layer to output of sixth layer.

The coverage is calculated as the ratio of the *receptive field* and the *input image Size* (Cao, 2015)

$$Coverage = \frac{\text{receptive field}}{\text{image size}} \tag{4}$$

As with capacity, the coverage of the network is calculated from the maximum available receptive field, i.e. the main branch of the network. For this network, the convolutional filters covered 66.4% of the input image and thus never covered more than the size of the image.

3.8.3. *Final architecture and training*

The network took as input 128 × 128 RGB images and outputted a vector for each image with one entry for each class as illustrated in Fig. 2. The network had one 5 × 5 convolutional layer, followed by a 2 × 2 max-pooling layer. This was mapped into a 1 × 1 convolutional layer, which decreased the number of filters from 64 to 32. Next came two residual blocks, each consisting of two 5 × 5 convolutional layers that was followed by a 2 × 2 max-pooling layer. Afters this, there was one more 5 × 5 convolutional layer and 2 × 2 max-pooling layer. Finally the network had two fully connected layers, which produced a vector with one entry for each of the 22 classes. The entry with the highest value determined the predicted class of the plant.

After the convolutional layers and after the residual blocks, each batch was normalised to its standard deviation. In total, the network had 1,218,614 learnable parameters, which is rather small compared to e.g. the 60 M parameters of Alexnet (Krizhevsky et al., 2012).

The parameters of the first convolutional layer were initialised to the weights from the VGG16 network (Simonyan & Zisserman, 2014) that was trained on the ImageNet dataset. Reusing these weights makes sense, as the first layers of the network contain the less abstract features, i.e. edge and blob detectors.

These features can therefore be reused even though they are learned from a completely different dataset. The parameters of the remaining layers were initialized randomly by using Glorot and Bengio's initialization, where weights are sampled from a normal distribution, scaled relative to the number of inputs and outputs from a layer. During the training, a 50% dropout was introduced before the two fully-connected layers. This worked as a regularisation that generates more robust features as the neurons are prevented from relying on other neurons. The network was trained using mini-batches with 200 images per batch in order to speed up the gradient update. Mini-batch training makes the gradient less precise, as it is only based on few samples. But, on the other hand, it increases the convergence rate. Furthermore, Mini-batch training helps in keeping the memory usage low. Implementation was made using the Theano-based Lasagne library for Python (Dieleman et al., 2015).

4. Results and discussion

Figure 4 shows the loss and accuracy for each epoch of the training. In order to achieve the highest accuracy possible without over-fitting the network, the training was stopped after 18 epochs. At this point the classification accuracy of the test set was 86.20%. After the 18th epoch, the validation loss starts to flatten and the gap between the training and validation loss increases. From the confusion matrix in Fig. 5, the fraction of misclassifications for each of the 22 species is

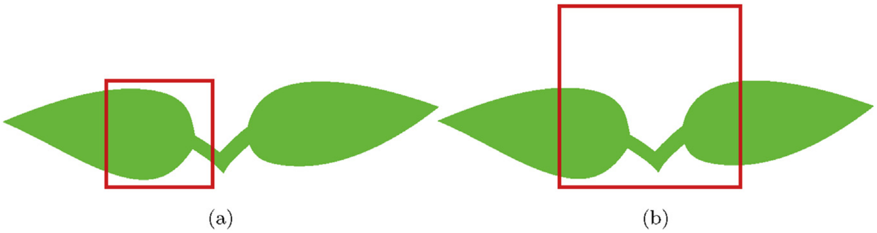


Fig. 3 – (a) A filter with a low capacity is only able to detect local features in the input image whereas (b) a filter with a high capacity is able to detect complex structures, such as multiple leaves. The red squares illustrate the convolutional filters.

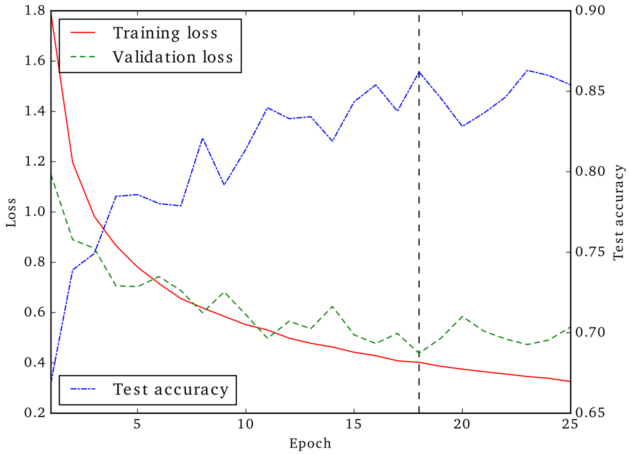
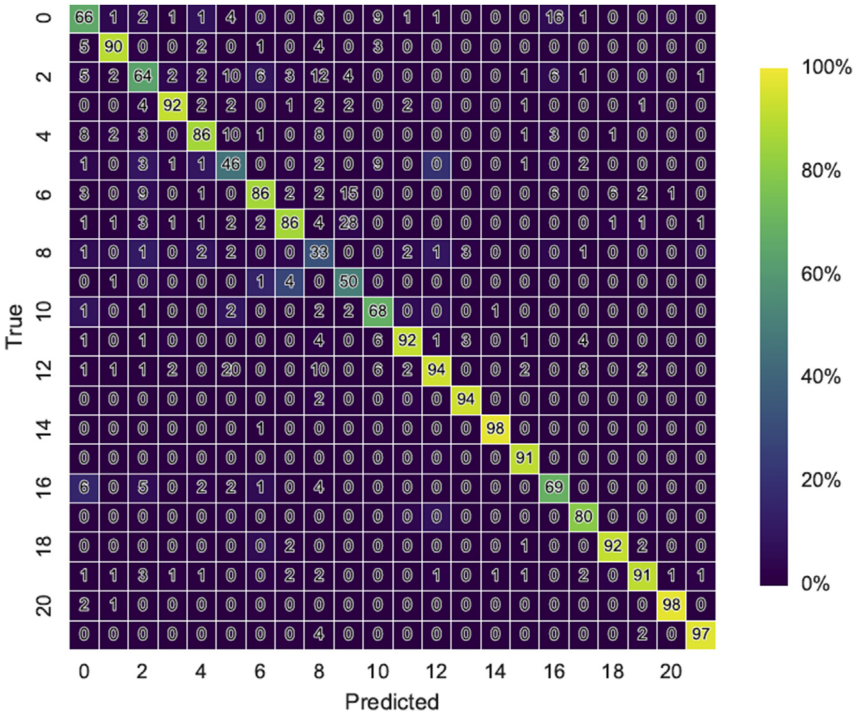


Fig. 4 – Cross entropy loss and classification accuracy. The vertical line at epoch 18 shows where the training was stopped.

shown. Here it is seen that Thale Cress (#14 *Arabidopsis thaliana*), Sugar Beet (#20 *Beta vulgaris*) and Barley (#21 *Hordeum vulgare* L.) were often correctly classified, with an accuracy of 98%, 98% and 97% respectively. Whereas Veronica (#5Veronica), Field Pancy (#8 *Viola arvensis*) and Broadleaved grasses (#9 *Poaceae*) were often misclassified. Of these three species, only 46%, 33% and 50% were classified correctly. Veronica (#5) was often confused with plants in the cabbage family (#12) and Broad-leaved grasses (#9) are often confused with Narrow-leaved grasses (#7). However, there was no clear species that Field Pansy (#8) was confused with. The classification accuracies for these three species were, however, still well above random assignment.

Overall, the classes with the most species were also the classes with the highest classification accuracies. This is because the aim of the training was to get the most plants classified correctly, without taking into account how these plants are distributed among the 22 classes. Therefore, classes with few image samples contributed less to the overall loss.



	0		2		4		6		8		10		12		14		16		18		20	
0	66	1	2	1	1	4	0	0	6	0	9	1	1	0	0	0	16	1	0	0	0	0
	5	90	0	0	2	0	1	0	4	0	3	0	0	0	0	0	0	0	0	0	0	0
2	5	2	64	2	2	10	6	3	12	4	0	0	0	0	0	1	6	1	0	0	0	1
	0	0	4	92	2	2	0	1	2	2	0	2	0	0	0	1	0	0	0	1	0	0
4	8	2	3	0	86	10	1	0	8	0	0	0	0	0	0	1	3	0	1	0	0	0
	1	0	3	1	1	46	0	0	2	0	9	0	0	0	0	1	0	2	0	0	0	0
6	3	0	9	0	1	0	86	2	2	15	0	0	0	0	0	0	6	0	6	2	1	0
	1	1	3	1	1	2	2	86	4	28	0	0	0	0	0	0	0	0	1	1	0	1
8	1	0	1	0	2	2	0	0	33	0	0	2	1	3	0	0	0	1	0	0	0	0
	0	1	0	0	0	0	1	4	0	50	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	1	0	0	2	0	0	2	2	68	0	0	0	1	0	0	0	0	0	0	0
	1	0	1	0	0	0	0	0	4	0	6	92	1	3	0	1	0	4	0	0	0	0
12	1	1	1	2	0	20	0	0	10	0	6	2	94	0	0	2	0	8	0	2	0	0
	0	0	0	0	0	0	0	0	2	0	0	0	0	94	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	98	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0
16	6	0	5	0	2	2	1	0	4	0	0	0	0	0	0	0	69	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0
18	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	92	2	0	0
	1	1	3	1	1	0	0	2	2	0	0	0	1	0	1	1	0	2	0	91	1	1
20	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	0
	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	2	0	97

Fig. 5 – Confusion matrix for the prediction of the 21 species listed in Table 3. The numbers indicates the percentage of correct classifications for the given class.

The network performance for Veronica (*Veronica*), Field Pancy (*V. arvensis*) and Broad-leaved grasses (*Poaceae*) were therefore expected to be better if more training data was available.

Table 4 shows the results obtained in this study, compared with other studies that have previously classified seedlings using computer vision. The classification accuracy from this study is on par with some of the studies when looking at the mean classification accuracy. In contrast to the previous studies, the images used in this study come from different data sources that contain far more different species than in the other previous studies. At the same time the plants in these images are present at multiple different growth stages, which further increases the complexity.

A direct comparison with all of these methods is, however, not possible. The method used by Søgaard (2005), Giselsson (2010) and Dyrmann and Christiansen (2014) all rely on features that are unable to handle a segmentation that causes some plants to be split in multiple parts. Åstrand and Baerveldt (2002) used only colour features for distinguishing plants. As our dataset was composed of images from different datasets, containing different illumination and camera settings, the method by Åstrand and Baerveldt (2002) is therefore not applicable to this dataset. Instead, the methods described by Kazmi (2014) and Golzarian and Fricka (2011) have been implemented and tested against our method using the same dataset. Kazmi (2014) and Golzarian and Fricka (2011) also segmented plant material from soil prior to the classification. However, their methods do not fail if plants are not kept as a single connected component. When using the Bag of Visual Words approach by Kazmi (2014) with Hessian-Laplace based SIFT features, an average accuracy of 42.5% is achieved for the images of 22 plants species in the present study. Originally Golzarian and Fricka (2011) used the assumption that the principal components of features for each plant species are normal distributed. The decision boundaries for a linear classifier were then set based on confidence intervals of these distribution. As far more species and the same number of principal components are used here, the distributions of principal components for each species overlap each other, thus the approach for determining decision boundaries is not feasible. Instead a linear classifier has been trained to determine the decision boundaries. By using that approach, an accuracy of 12.2% was achieved. These accuracies should be compared to the classification accuracy of 86.2% achieved using our method.

4.1. *Implementation in operational setup*

This network is intended as a support to the farmer when he is to decide which herbicides, he must apply to his field. The farmer can thus collect photos from the field and the network can estimate the weed population, which can be used as a basis for determining the optimal herbicide composition. When using mechanical weed control, it is only necessary to be able to recognize plants as either weeds or crop. As the present system can handle 22 classes, it will therefore be beyond what is needed for that role, although this neural network could also be trained only to separate the two classes.

In an operational setup, the intra-variance for each species can be lowered, as the camera will be fixed. Fixing the camera will ensure that the scaling of 305 plants is known, which is expected to be a useful feature for plant recognition. Furthermore, shading and artificial lightning can ensure that the conditions for the images are the same for all images, which will make segmentation unnecessary. Omitting the segmentation should also increase the classification accuracy further as stem regions, that are removed by the segmentation, will be kept.

Currently, the average processing time for one image is 27 ms on a Nvidia Quadro 2000M. However, this time is only for classifying the plants. The calculation time for localising the plants should therefore be added to this value before it can be determined whether this speed is acceptable, or not, for real-time weed control applications.

5. Conclusion

A convolutional neural networks was constructed for distinguishing images of seedlings at early growth stages. The study was conducted on vertically photographed images of seedlings covering 22 different plants species or families, which is believed to be a state-of-the-art in terms of number of different species. The classification accuracy of the network ranged from 33% up to 98% with an average accuracy of 86.2% Especially Thale Cress (*A. thaliana*), Sugar Beet (*B. vulgaris*) and Barley (*H. vulgare* L.) were often correctly classified, with accuracies of 98%, 98% and 97% respectively, whereas the network had problems in classifying Veronica (*Veronica*), Field Pancy (Viola *arvensis*) and Broad-leaved grasses (*Poacher*). This problem is believed to be due to a low number of training samples for these species.

Table 4 – Comparison of classification accuracies from previous seedling classification studies. Last column shows the classification accuracies when the methods are trained and tested on our dataset with 22 classes.

Method	Number of species/groups	Total number of samples (test and train)	Original classification accuracy	Classification accuracy on our 22 classes
Søgaard (2005)	3	93	65%–93%	–
Åstrand and Baerveldt (2002)	2	587	91%	–
Giselsson (2010)	8	1698	94.8%	–
Dyrmann and Christiansen (2014)	7	2436	95.8%	–
Kazmi (2014)	2	474	99%	42.5%
Golzarian and Fricka (2011)	3	286	82.4%–88.2%	12.2%
Current study	22	10,413	86.2%	86.2%

Acknowledgements

The work was founded by Green Development and Demonstration Programme (GUDP) under the Danish Ministry for Food, Agriculture and Fisheries.

REFERENCES

Aarhus University - Department of Agroecology, & SEGES. (2015). *Ukrudtsnøglen*. plantevaernonline.dlbr.dk.

Åstrand, B., & Baerveldt, A. J. (2002). An agricultural mobile robot with vision based perception for mechanical weed control. *Autonomous Robots, 13*, 21–35. http://link.springer.com/article/10.1023/A:1015674004201.

Cao, X. (2015). *A practical theory for designing very deep convolutional neural networks Classifier Level*.

Dieleman, S., Schlüter, J., Raffel, C., Olson, E., Sønderby, S. K., Nouri, D., et al. (2015). *Lasagne: First release*. http://dx.doi.org/10.5281/zenodo.27878.

Dyrmann, M., & Christiansen, P. (2014). *Automated classification of seedlings using computer vision*. *Tech. Rep*. Aarhus: Aarhus University. http://plant_recognition.sdu.dk/files/ACSUCV2015.pdf.

Giselsson, T. M. (2010). *Real time crops and weeds classification from top down images of plant canopies*. University of Southern Denmark. http://thomasg.dk/files/masterThesis.pdf.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feed-forward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), 9* pp. 249–256). http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2010_GlorotB10.pdf.

Golzarian, M. R., & Frick, R. A. (2011). Classification of images of wheat, ryegrass and brome grass species at early growth stages using principal component analysis. *Plant Methods, 7*(1), 28. http://www.plantmethods.com/content/7/1/28.

He, K., Zhang, X., Ren, S., & Sun, J. (dec 2015). *Deep residual learning for image recognition* (Vol. 7) (3) http://arxiv.org/abs/1512.03385.

Hodgson, J. M. (1968). *The nature, ecology, and control of Canada thistle*. Technical Bulletins 171614. United States Department of Agriculture, Economic Research Service http://EconPapers.repec.org/RePEc:ags:uerstb:171614.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. Arxiv. http://arxiv.org/abs/1502.03167.

Jørgensen, L. N., Noe, E., Langvad, A.-m., Rydahl, P., Jensen, J. E., Ørum, J. E., et al. (2007). *Vurdering af Planteværn Onlines økonomiske og miljømæssige effekt* (Assessment of Crop Protection Online's economic and environmental effect). 115. Bekæmpelsesmiddelforskning fra Miljøstyrelsen.

Kazmi, W. A. (2014). *Computer vision based weed detection and 3D plant imaging*. Aalborg University. Ph.D. thesis.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1–9.

Meier, U. (2001). *Growth stages of mono-and dicotyledonous plants – BBCH monograph*. Federal Biological Research Centre for Agriculture and Forestry. http://www.bba.de/veroeff/bbch/bbcheng.pdf.

Minervini, M., Abdelsamea, M. M., & Tsaftaris, S. A. (2014). Image-based plant phenotyping with incremental learning and active contours. *Ecological Informatics, 23*, 35–48 (Special Issue on Multimedia in Ecology and Environment) http://doi.org/10.1016/j.ecoinf.2013.07.004.

Oerke, E.-C. (2006). Crop losses to pests. *The Journal of Agricultural Science, 144*(01), 31.

RoboWeedSupport. (2015). *Roboweedsupport.com*. http://roboweedsupport.com/.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large Scale visual recognition challenge. *International Journal of Computer Vision (IJCV), 115*(3), 211–252.

Scharr, H., Minervini, M., Fischbach, A., & Tsaftaris, S. A. (2014). Annotated image datasets of rosette plants. *Forschungszentrum Julich* (Technical Report No. FZJ-2014–03837).

Simonyan, K., & Zisserman, A. (sep 2014). Very deep convolutional networks for large-scale image recognition. *arXiv*, 1–13. http://arxiv.org/abs/1409.1556.

Slaughter, D. C., Giles, D. K., Fennimore, S. A., & Smith, R. F. (2008). Multispectral machine vision identification of lettuce and weed seedlings for automated weed control. *Weed Technology, 22*(2), 378–384. http://www.bioone.org/doi/abs/10.1614/WT-07-104.1.

Søgaard, H. T. (jul 2005). Weed classification by active shape models. *Biosystems Engineering, 91*(3), 271–281. http://linkinghub.elsevier.com/retrieve/pii/S1537511005000772.

Woebbecke, D. M., Meyer, G. E., Bargen, K. V., & Mortensen, D. A. (1995). Color indices for weed identification under various soil, residue, and lighting conditions. *Transactions of the ASAE, 38*(1), 259–269.