Available online at www.sciencedirect.com

ScienceDirect

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

Research Paper

# Transfer learning for the classification of sugar beet and volunteer potato under field conditions

*Hyun K. Suh*[*], *Joris IJsselmuiden, Jan Willem Hofstee, Eldert J. van Henten*

*Farm Technology Group, Wageningen University, P.O.box 16, 6700 AA, Wageningen, the Netherlands*

ARTICLE INFO

*Article history:*
Received 23 December 2016
Received in revised form
11 June 2018
Accepted 19 June 2018
Published online 9 July 2018

*Keywords:*
Automated weed control
Weed classification
Deep learning
Transfer learning
Convolutional neural network

Classification of weeds amongst cash crops is a core procedure in automated weed control. Addressing volunteer potato control in sugar beets, in the EU Smartbot project the aim was to control more than 95% of volunteer potatoes and ensure less than 5% of undesired control of sugar beet plants. A promising way to meet these requirements is deep learning. Training an entire network from scratch, however, requires a large dataset and a substantial amount of time. In this situation, transfer learning can be a promising solution. This study first evaluates a transfer learning procedure with three different implementations of AlexNet and then assesses the performance difference amongst the six network architectures: AlexNet, VGG-19, GoogLeNet, ResNet-50, ResNet-101 and Inception-v3. All nets had been pre-trained on the ImageNet Dataset. These nets were used to classify sugar beet and volunteer potato images taken under ambient varying light conditions in agricultural environments. The highest classification accuracy for different implementations of AlexNet was 98.0%, obtained with an AlexNet architecture modified to generate binary output. Comparing different networks, the highest classification accuracy 98.7%, obtained with VGG-19 modified to generate binary output. Transfer learning proved to be effective and showed robust performance with plant images acquired in different periods of the various years on two types of soils. All scenarios and pre-trained networks were feasible for real-time applications (classification time < 0.1 s). Classification is only one step in weed detection, and a complete pipeline for weed detection may potentially reduce the overall performance.

## 1. Introduction

Volunteer potato is a source of potato blight (*Phytophthora infestans*) and viral diseases. Volunteer potato in a sugar beet field can reduce the crop yield by 30% (O'Keeffe, 1980). There is a statutory obligation for sugar beet farmers in the Netherlands to control volunteer potato plants to no more than two remaining plants per m$^2$ by 1st of July (Nieuwenhuizen, 2009). For the automated control of volunteer potato in a sugar beet field, a vision-based and small-sized robot was developed within the EU-funded project SmartBot. Due to the small size of the robot and the required battery operation, the platform design had to refrain from additional infrastructure and needed to be able to robustly detect weeds in a scene that was

[*] *Corresponding author.*
E-mail address: davidsuh79@gmail.com (H.K. Suh).

https://doi.org/10.1016/j.biosystemseng.2018.06.017

fully exposed to ambient lighting conditions. Additional infrastructure such as a hood and lighting equipment, as used for instance by Nieuwenhuizen, Hofstee, and Van Henten (2010) and Lottes et al. (2016), was not considered viable. The robotic platform is shown in Fig. 1.

The classification of weeds amongst cash crops, i.e. weed/crop discrimination, is the core procedure for automated weed detection. In a pipeline for weed detection, vegetation segmentation is followed by classification of the segmented vegetation into weeds and crop. This classification step traditionally involves two aspects: selection of the discriminative features as well as selection of the classification techniques (Suh, Hofstee, IJsselmuiden, & Van Henten, 2016).

Regarding the features used for discrimination, many studies have used colour, shape (biological morphology) and texture on an individual basis or as a combination of multiple features (Ahmed, Al-Mamun, Bari, Hossain, & Kwan, 2012; Gebhardt & Kühbauch, 2007; Persson & Åstrand, 2008; Pérez, López, Benlloch, & Christensen, 2000; Slaughter, Giles, & Downey, 2008; Swain, Nørremark, Jørgensen, Midtiby, & Green, 2011; Zhang, Kodagoda, Ruiz, Katupitiya, & Dissanayake, 2010; Åstrand & Baerveldt, 2002). However, these features have shown poor performance under widely varying natural light conditions (Suh, Hofstee, IJsselmuiden, & Van Henten, 2018). Other features such as Scale Invariant Feature Transform (SIFT) (Lowe, 2004) and Speeded Up Robust Features (SURF) (Bay, Ess, Tuytelaars, & Van Gool, 2008) have shown their potential in recent studies in the classification of plant species (Kazmi, Garcia-Ruiz, Nielsen, Rasmussen, & Andersen, 2015; Suh et al., 2018; Wilf et al., 2016). However, the highest classification accuracy using SIFT and SURF obtained in Suh et al. (2018) is still not a satisfactory performance in view of the requirements set by the previous study of Nieuwenhuizen (2009): the resulting automatic weeding system should effectively control more than 95% of the volunteer potatoes as well as ensure less than 5% of undesired control of the sugar beet plants. Therefore, within the framework of the EU Smartbot Project, a solution was needed that achieves a classification accuracy of 95% or more as well as a misclassification of both sugar beet [false-negative (FN)] and volunteer potato [false-positive (FP)] of less than 5%. In addition, a classification time of less than 0.1 s per image was also needed because these algorithms should be used in a real-time field application.

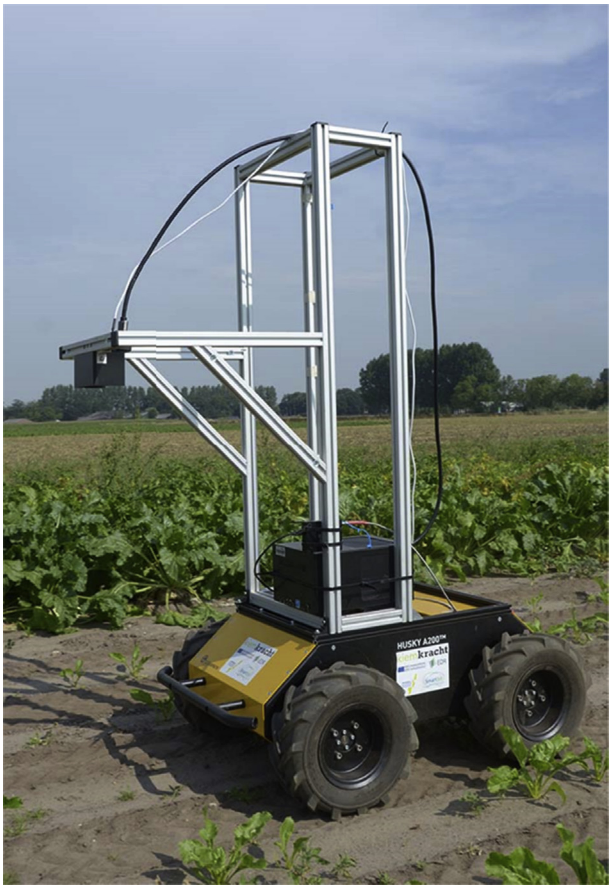

**Fig. 1 – The robotic platform for volunteer potato control in a sugar beet field.**

A promising way to meet these requirements is to use a deep learning approach. In recent studies, the deep neural network has shown its potential in an agricultural context for plant identification and classification. Grinblat, Uzal, Larese, and Granitto (2016) used a convolutional neural network (ConvNet, or CNN), a specific type of deep network, for plant identification from leaf vein patterns. Although the binary images of vein patterns were used, the study showed the potential of ConvNet for plant identification. Sun, Liu, Wang, and Zhang (2017) used a residual network (ResNet), one of the most common ConvNet architectures used for classification tasks, for plant species identification with images acquired by mobile phones. A 91.78% of classification accuracy was obtained, but they needed 10,000 images to train the network. Dyrmann, Karstoft, and Midtiby (2016) classified 22 plants species using a ConvNet and obtained 86.2% of classification accuracy. In their study, images were acquired under controlled conditions, a quite distinct difference from the conditions that confronted SmartBot, and the number of images needed to train the network from scratch was even more than 10,000. Obtaining such a large number of images, however, is a challenging task in agricultural fields (Xie, Jean, Burke, Lobell, & Ermon, 2016). Besides, training an entire ConvNet from scratch requires a substantial amount of time (Jean et al., 2016; Yosinski, Clune, Bengio, & Lipson, 2014) and is an expensive task that may be hard to realise in practice. Then, transfer learning can be a promising solution.

The objective and novelty of this paper are to deal with crop/weed classification under uncontrolled agricultural environments as well as to reduce the amount of data and time using transfer learning.

Transfer learning has gained its success in real-world applications (Jean et al., 2016; Shin et al., 2016; Yi; Sun, Wang, & Tang, 2014; Xie et al., 2016). Transfer learning, according to Goodfellow, Bengio, and Courville (2016), refers to exploiting what has been learned from one setting into another different setting. In transfer learning, the base network is trained on a base dataset and task, and then the (pre-)trained network is reused for another task (Yosinski et al., 2014). Interestingly enough, though the ConvNet is trained with a specific dataset to perform a specific task, the generic features extracted from ConvNet seem to be powerful and perform very well on other classification tasks as well (Donahue et al., 2014; Razavian, Azizpour, Sullivan, & Carlsson, 2014). Transfer learning has recently been applied in several agricultural applications such as disease detection (Fuentes, Yoon, Kim, & Park, 2017); however, the transfer learning procedure has not yet been investigated in detail in plant classification.

In this study, firstly, three different transfer learning scenarios were evaluated using AlexNet (Krizhevsky, Sutskever, & Hinton, 2012). Then, the performance of the following six pre-trained networks was compared: AlexNet, VGG-19 (Simonyan & Zisserman, 2015), GoogLeNet (Szegedy, Liu et al., 2015), ResNet-50 and ResNet-101 (He, Zhang, Ren, & Sun, 2016a), and Inception-v3 (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2015). The classification performance in both evaluations was analysed regarding classification accuracy as well as training and classification time, given the fact that this approach should be used in a real-time field application.

The first section of this paper describes ConvNets and their popular architectures. The following section, describes three different scenarios in transfer learning. Then the performance assessment amongst six pre-trained networks is described. The experimental setup including field image dataset collection and the performance measures to be used are described. Then, the experimental results are shown with the corresponding discussions, leading to the conclusions.

## 2. Convolutional neural networks and popular architectures

Convolutional neural networks (ConvNets, or CNNs) are a specialised type of deep neural networks that are designed to process multi-dimensional data such as signals (1D), images (2D) and videos (3D) (LeCun, Bengio, & Hinton, 2015; LeCun, Bottou, Bengio, & Haffner, 1998). ConvNets have gained huge success in many applications since AlexNet won the ImageNet competition in 2012 with a breakthrough performance (Sainath, Mohamed, Kingsbury, & Ramabhadran, 2013; Schwing & Urtasun, 2015; Sermanet et al., 2013; Zeiler & Fergus, 2014). Motivated by the success of AlexNet, further deep ConvNets were proposed in the recent literature such as VGG-19, GoogLeNet, ResNet and Inception-v3. These ConvNets contain from several to hundred layers of convolutions with non-linear activation functions, such as Sigmoid, Tanh, and ReLU (Rectified Linear Units), applied to the results. A different set of convolution filters is applied over each layer, and then the output of the convolutions are combined to maintain the local connectivity between neurons of adjacent layers. This local connectivity enables each neuron to be connected only to a small local subset of the given image which helps to reduce the number of parameters in the whole network as well as to make the computation more efficient (Chen, Wu, Fan, Sun, & Naoi, 2014). Such a deep layered ConvNet structure enables the network to learn the best features during the training process automatically and will in most cases outperform hand-crafted feature extractors which generally require an extensive engineering skill and knowledge (Hu, Xia, Hu, & Zhang, 2015; LeCun et al., 2015).

AlexNet, one of the first ConvNets, contains seven layers besides input and output layers (Fig. 2). The first five layers are convolutional layers (Conv layers) each followed by ReLU and max-pooling, which are non-linear activation and down-sampling functions to enhance the training time efficiency. The last three layers are fully-connected layers (FC layers) composed of two FC layers each with a 4096-dimensional activation vector followed by one FC layer (softmax layer) with 1000 activation neurons, thus producing a classification score in terms of 1000 different categories.

In VGG-19, only 3 × 3 filters are used in all convolutional layers to reduce the number of parameters in the network. Furthermore, the use of max pooling between convolutional layers largely reduces the network volume (Simonyan & Zisserman, 2015). Like AlexNet, the last layers are two FC layers, each with a 4096-dimensional activation vector, followed by a softmax layer.

In GoogLeNet, the Inception Module was introduced to process the required operations in parallel. The Inception Module acts as an efficient multi-level feature extractor and makes the network considerably smaller and faster. Szegedy, Liu et al. (2015) and Szegedy, Vanhoucke et al., (2015) reported that GoogLeNet was smaller and faster than VGG-19 even though GoogLeNet contained more layers (22 layers) than VGG-19 (19 layers).

ResNet (Residual Network) consists of several basic residual blocks which provide a shortcut connection between layers. This shortcut connection makes it possible to train hundreds or more layers while achieving enhanced performance. ResNet is primarily designed for large-scale data analysis and is developed with many different numbers of layers including 50 and 101 (Alom et al., 2018). ResNet-50 and ResNet-101 contain, respectively, 50 and 101 convolutional layers including one FC layer at the end of the network (He, Zhang, Ren, & Sun, 2016b).

Inception-v3 extends the original GoogLeNet implementation and enhances the Inception Module to improve the accuracy by factorisation of convolutions and improved normalisation (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017; Szegedy, Vanhoucke, et al., 2015). V3 simply indicates that this network is the 3rd version, updated and released by Google.

All these networks are available as pre-trained ConvNets which have been trained on ImageNet Dataset[1] to classify 1000 object categories such as desk, chair, keyboard, animals, etc. These ConvNets are used as pre-trained networks in this paper.

## 3. Three scenarios for transfer learning

Transfer learning aims to overcome the shortage of training data and time by transferring information or features that are extracted from the pre-trained ConvNets (Oquab, Bottou, Laptev, & Sivic, 2014; Weiss, Khoshgoftaar, & Wang, 2016). AlexNet was used as a pre-trained ConvNet. Two options are

[1] The ImageNet Dataset contains 1.2 million labelled training images and 50,000 test images, with each image labelled with one of 1000 classes (Yosinski et al., 2014).

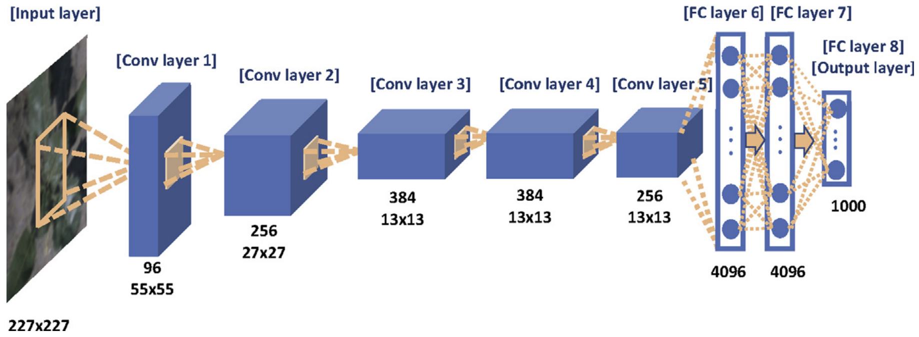

**Fig. 2 – The overall structure of AlexNet. The network is composed of five convolutional layers (Conv layer 1–5) and three FC layers (FC layers 6–8). FC layer 6 and 7 produce a 4096-dimensional activation vector. The last layer, FC layer 8, is the output layer which produces classification scores on 1000 categories as the AlexNet was originally designed to classify 1000 different classes. The output size of each layer changes as a convolution process is being applied (Conv layer 1–5).**

available in transfer learning: use of ConvNet as a feature extractor and use of ConvNet as a classifier. Based on these available options, three scenarios for transfer learning were formulated based on the following hypotheses:

1) Scenario 1: In this scenario, the hypothesis was tested whether, with or without retraining AlexNet, a classification accuracy of 95% or more could be achieved using the features extracted from FC6 and FC7 and using conventional classifiers.
2) Scenario 2: AlexNet was modified to produce binary classification output (i.e. sugar beet or volunteer potato). Once AlexNet was modified, it was fine-tuned with training images of sugar beet and volunteer potato. In this case, the hypothesis was that using more training data would lead to a better classification accuracy than the one obtained in scenario 1.
3) Scenario 3: Once AlexNet was modified and fine-tuned as in scenario 2, the hypothesis was that an improved classification accuracy might be achieved using the features extracted from FC6 and FC7 and using a conventional classification scheme as used in scenario 1.

A total of 1100 labelled plant images was used. Each plant image was resized to 227 × 227 pixels (RGB) using a default image resizing function in Matlab, as AlexNet has a predefined 227 × 227 pixel input size. No data augmentation was applied. In all scenarios, the classification performance was averaged over ten repetitions. The classifiers in scenario 1 and 3 were validated by 10-fold random cross-validation over ten repetitions on a separate set of random images.

To get more insight into performance differences amongst different classifiers for scenario 1 and 3, the Support Vector Machine (SVM), random forest and linear discriminant analysis (LDA) were used for classification. These classifiers have been used in many agricultural applications (Ahmed et al., 2012; Longchamps, Panneton, Samson, Leroux, & Thériault, 2009; Lottes et al., 2016; Zhang et al., 2010), but it was not known *a priori* which classifier performs best on the classification task in hand. In the SVM, three different polynomial kernels (linear, quadratic and cubic) were evaluated.

### 3.1. *Scenario 1 – AlexNet as a fixed feature extractor*

One of the transfer learning approaches is to use a pre-trained ConvNet as a feature extractor (Jean et al., 2016). Without retraining the whole network, the features extracted from the last layers in ConvNets can be used as a feature vector which has generic properties applicable to other tasks using a conventional classification scheme (Donahue et al., 2014; Gong, Jia, Leung, Toshev, & Ioffe, 2014). In this scenario, the 4096-dimensional feature vector was extracted from each of the last two FC layers, FC layer 6 (FC6) and FC layer 7 (FC7) as AlexNet yields vectors with 4096 feature values in these last layers. To investigate the difference in classification performance between the two layers of FC6 and FC7 in AlexNet, the extracted features in FC6 and FC7 were used individually to train and validate the following classifiers: SVM (with three different kernels), random forest, and LDA (Fig. 3). The flowchart of scenario 1 is shown in Fig. 4.

### 3.2. *Scenario 2 – Modified and fine-tuned AlexNet as a binary classifier*

Inspired by Papadomanolaki, Vakalopoulou, Zagoruyko, and Karantzalos (2016) who modified a pre-trained ConvNet to classify satellite data, in this scenario AlexNet was modified to generate a binary classification output: sugar beet or volunteer potato. The original AlexNet was designed to classify 1000 objects, having 1000 activation neurons in the last layer of the network (Krizhevsky et al., 2012). This last layer in the original AlexNet was removed, and two new fully-connected layers (FC layers 8′ and 9′) were added (Fig. 5). The modification details are as follows:

1) The last FC layer in AlexNet was removed (FC layer 8 in Fig. 2).
2) A new FC layer with 64 neurons (square root of 4096) was added to the end, followed by a ReLU (Rectified Linear Unit), as ReLU was applied to the output of every layer in the original AlexNet.
3) A new FC layer with two neurons was added with a 2-way softmax to produce the binary classification output.

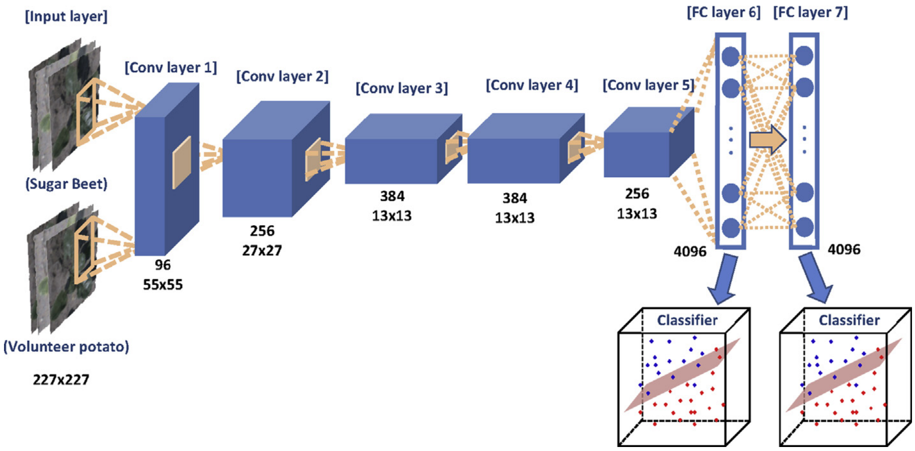

Fig. 3 – Scenario 1 – AlexNet as a feature extractor. Two FC layers, FC layer 6 and 7, were each used individually as a feature extractor. For the classification of sugar beet and volunteer potato, three classifiers were evaluated: SVM (with linear, quadratic, and cubic kernels), random forest, and LDA.

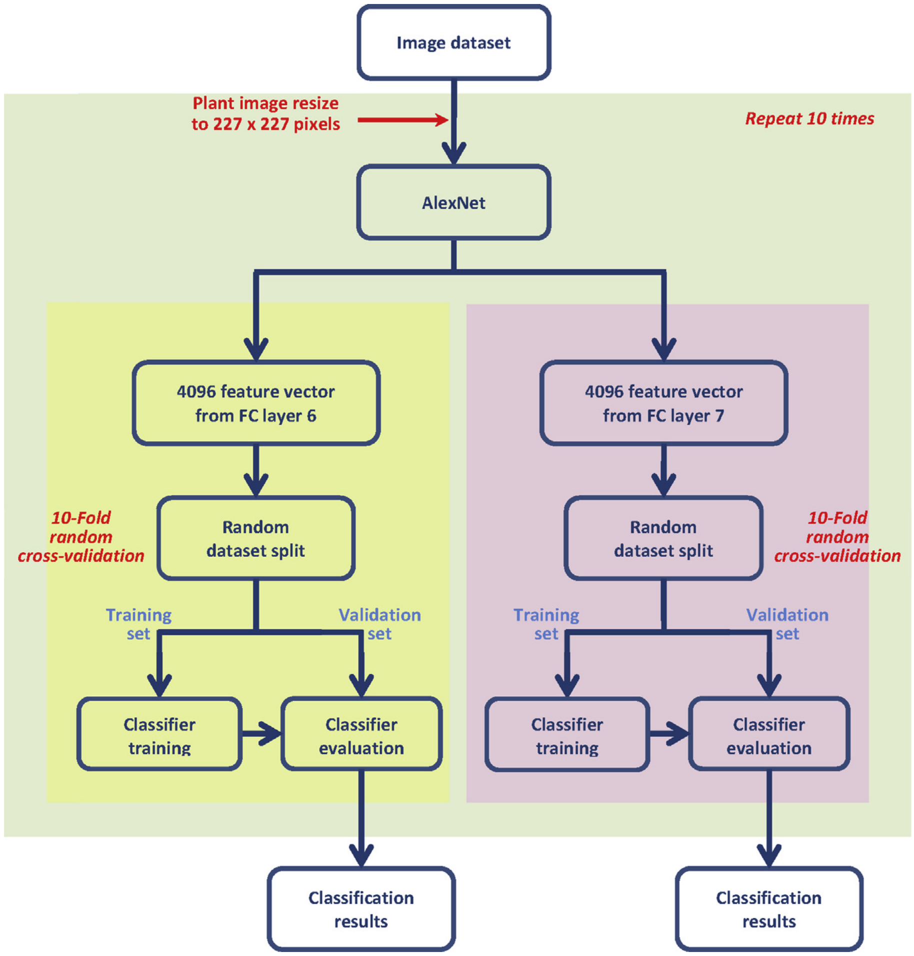

Fig. 4 – Flowchart of scenario 1. A total of 1100 labelled plant images was used. Each plant image was resized to 227 × 227 pixels (RGB). The classifier results were validated by 10-fold random cross-validation over ten repetitions.

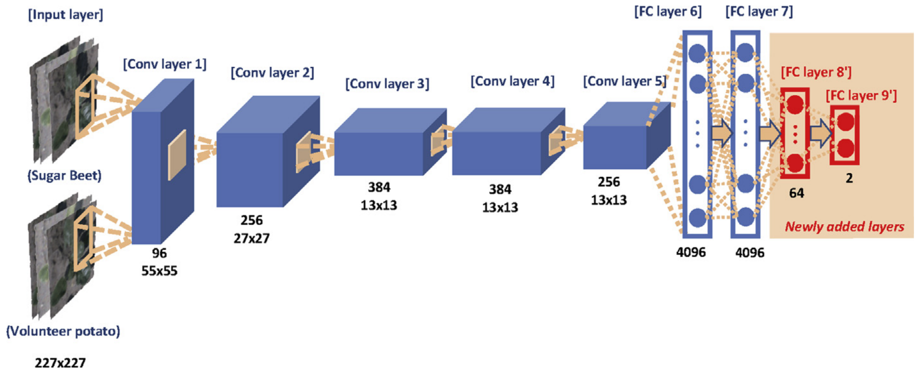

**Fig. 5 – Scenario 2 – AlexNet was modified to generate a binary classification output: sugar beet or volunteer potato. The original FC layer 8 was removed, and new FC layers 8′ and 9′ were added in the end for AlexNet to generate a binary classification into sugar beet and volunteer potatoes.**

Between FC layer 7 (size of 4096 neurons) and FC layer 9' (size of two neurons for binary output), FC layer 8′ with 64 neurons was added to help smooth the dimensional reduction from 4096 to two (Fig. 5). Preliminary experimentation showed that this addition produced slightly better performance compared to having no layer in between.

The modified AlexNet was then fine-tuned on our image dataset, which had been acquired during three different periods of three different years on two different soil types. The modified AlexNet was trained using a stochastic gradient descent (SGD) method with a batch size of 128 examples and a momentum of 0.9 as these parameters had also been used for training the original AlexNet (Krizhevsky et al., 2012). In order not to change the parameters of original convolutional layers too much, the learning rate was fixed to 0.001, and the learning was stopped after 20 epochs.

The classification performance of the fine-tuned network was expected to depend on the number of training images used. The classification performance was evaluated while varying the number of training images from 200 to 900 with an increment of 100, in order to find the optimal number of training images needed for fine-tuning the AlexNet. Training images were randomly selected out of the 1100 available images in the dataset, and from the remaining images, 200 images were randomly selected for validation.

### 3.3. *Scenario 3 – Modified and fine-tuned AlexNet as a fixed feature extractor*

This scenario is a combination of scenario 1 and 2. The original AlexNet was first modified to generate a binary classification output (sugar beet or volunteer potato) as described in scenario 2. The modified AlexNet was fine-tuned with two different numbers of images, 300 and 800, which were randomly selected out of the 1100 plant images in the dataset. Then, new classifiers including SVM, random forest, and LDA were trained and validated using features extracted from the FC layers 6 and 7 (FC6 and FC7) separately as described in scenario 1. For classifier training and validation, 300 images were randomly selected from the remaining images.

During the fine-tuning process, the modified AlexNet was expected to adjust its node weights based on the given plant images. This change would alter the value of the activation vectors in (all) layers in the network, which was likely to improve the classification performance compared to scenario 1.

## 4. Classification performance amongst different ConvNet architectures

The following six pre-trained deep networks were evaluated to assess the classification performance amongst different ConvNet architectures: AlexNet, VGG-19, GoogLeNet, ResNet-50, ResNet-101 and Inception-v3. Each network was modified to produce binary classification output of sugar beet and volunteer potato, as was done in scenario 2 with AlexNet (Section 3.2), by removing the original last layer and adding two new FC layers. Then, each modified network was fine-tuned with 500 randomly selected images of sugar beet and volunteer potato: the remaining 600 images were used for validation. The number 500 was chosen for fine-tuning based on our preliminary studies as well as based on the fact that in scenario 2 with AlexNet, the accuracy improvement started to flatten after 500 (Fig. 8). Plant images were resized to correspond to the input size of each network. Unlike in the three scenarios (Subsections 3.1 to 3.3), data augmentation was applied here based on image transformations such as translation, rotation and flipping. Data augmentation includes a wide range of techniques used to generate new training images from the original ones by applying such random image transformations. Data augmentation is to increase the generalisability of the model, and in most cases leads to an improvement in classification accuracy. It can then be assessed if applying data augmentation here may yield a better performance compared to no data augmentation in scenario 2 above (in the case of AlexNet).

The training parameters here were the same as those described in Sections 3.1 to 3.3, but two different epochs, 20 and 30, were used for training to gain insight into the performance difference with different numbers of training epochs. No layers were frozen during training as, in our preliminary examination, it yielded slightly better performance than freezing some portion of the layers. The classification performance of each network was averaged over five repetitions.

## 5. Experimental setup

### 5.1. *Field image collection and image dataset*

For crop image acquisition, a camera was mounted at a height of 1 m and perpendicular to the ground on a custom-made frame carried by a mobile platform (Husky A200, Clearpath, Canada) (Fig. 6). The camera (NSC1005c, NIT, France) was equipped with two Kowa 5 mm lenses (LM5JC10M, Kowa, Japan) with a fixed aperture. The camera was set to operate in automatic acquisition mode with default settings. The camera had two identical complementary metal-oxide semiconductor (CMOS) sensors providing left and right images. Though this camera is intended to be used for stereovision, this feature was not used in this research. Left and right images were individually treated and separately used, each having image resolution of 1280 × 580 pixels. The area of ground covered was 1.3 m × 0.7 m per image (pair), corresponding to three crop rows of sugar beet. The acquisition program was implemented in LabVIEW (National Instruments, Austin, USA) to acquire five images per second. Raw format images (TIFF) were initially acquired in the field, and debayer was processed offline to convert the raw format image into RGB colour. Field images were taken while the mobile platform was manually controlled with a joystick and driven along crop rows using a controlled travelling speed of 0.5 m $s^{-1}$.

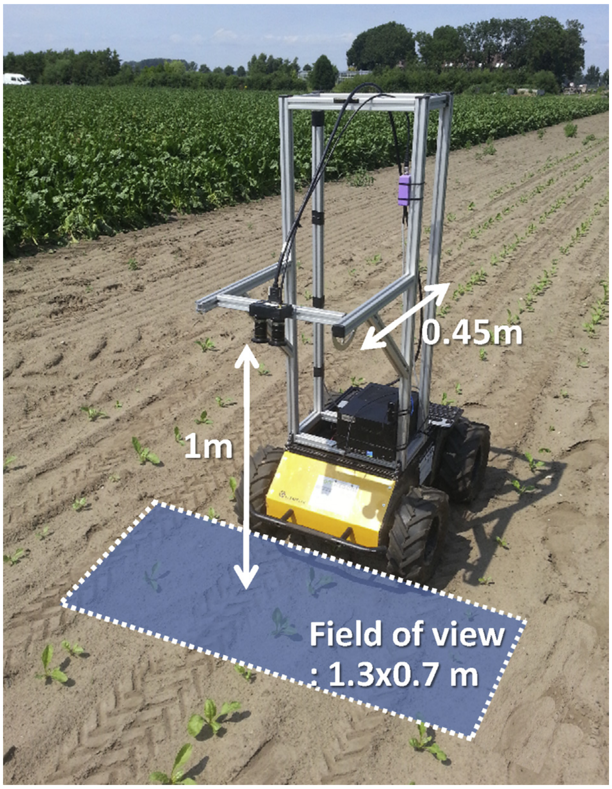

**Fig. 6 – Field images were acquired with a camera mounted at the height of 1 m viewing perpendicular to the ground surface resulting in a field of view of 1.3 × 0.7 m. A mobile platform, Clearpath Husky, was manually controlled with a joystick and driven along crop rows using a controlled travelling speed of 0.5 m $s^{-1}$.**

Sugar beet was sown three times (spring, summer, and autumn) each year in 2013, 2014 and 2015 in sandy and clay soil at Unifarm experimental sites in Wageningen, The Netherlands. One week after sowing the sugar beet, potatoes were planted in random locations throughout the fields. The plant images were acquired under a wide range of illumination and weather conditions for several days in June, August and October of 2013, in May, June, July and September of 2014 and in May, June, July and October of 2015.

For the labelled image dataset used in this study, a total of 1100 individual plant images was manually extracted from selected field images: 550 sugar beet plants and 550 volunteer potato plants. During the selection of this dataset, images with different ambient light conditions were included as well as images containing various stages of plant growth and shadows which were caused by neighbouring plants and/or the robotic platform. The size of each plant image in the dataset varied from 73 × 60 pixels to 310 × 315 pixels. Example images from this dataset are shown in Fig. 7.

### 5.2. *Software and hardware platform*

All procedures were implemented in Matlab (The MathWorks Inc, Natick, MA, USA) using the Neural Network Toolbox™, Statistics and Machine Learning Toolbox™ and MatConvNet toolbox (Vedaldi & Lenc, 2015). As computing hardware platforms, two cloud servers were used from Amazon Elastic Compute Cloud (EC2) and Paperspace GPU Cloud. Amazon EC2 was used in the scenarios (Subsections 3.1 to 3.3), and Paperspace GPU Cloud was used in the ConvNet comparison. These cloud servers provided a simple and easy setup of a high-performance computing platform with reduced cost of maintenance. Amazon EC2 was equipped with an Intel® Xeon® CPU E5-2670 2.5 GHz processor, 15 GB memory and Nvidia Grid™ K520 GPU running 64-bit Windows Server 2012. Paperspace GPU cloud was equipped with an Intel® Xeon® CPU E5-2623 2.6 GHz processor, 30 GB memory and Nvidia Quadro P5000 16GB GPU running 64-bit Ubuntu 16.04 LTS.

### 5.3. *Performance measures*

A binary classification was performed in this study: sugar beet or volunteer potato. The classification performance measures for this study are described below.

A confusion matrix (Table 1) was used to assess and compare the classification performance. The classification accuracy was calculated along with training and classification time, as this work is intended for real-time field application. The classification accuracy and training time were averaged over ten and five trials in the scenarios and the network comparison, respectively. The classification time measured was the time required to classify a single plant image on the cloud servers.

$$\text{Classification Accuracy} = \frac{TP + TN}{TP + FN + FP + FN} \tag{1}$$

where: TP is true-positive; FP is false-positive; TN is true-negative; FN is false-negative.

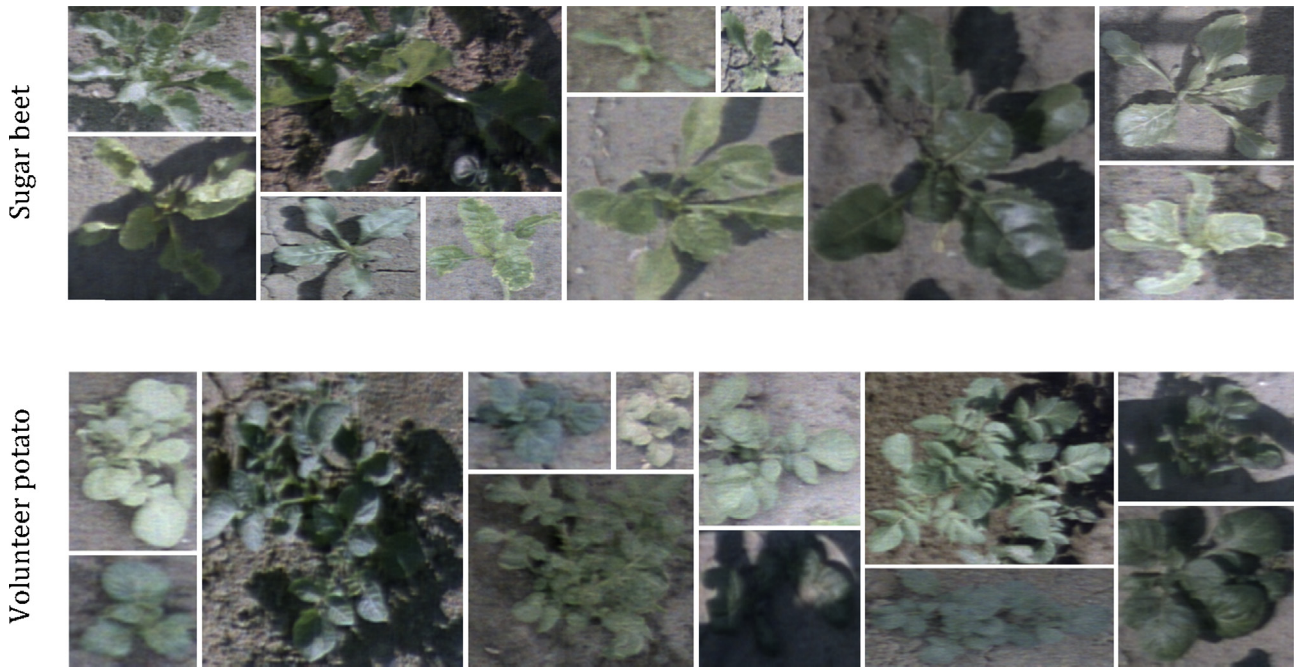

**Fig. 7 – Example images from the field image dataset containing a total of 1100 plant images with 550 sugar beet (top) and 550 volunteer potatoes (bottom). During the selection of this dataset, images with different ambient light conditions were included taken on both sandy and clay soils as well as images containing shadows and various stages of plant growth.**

## 6. Results

### 6.1. *Three scenarios for transfer learning*

6.1.1. *Scenario 1 – AlexNet as a fixed feature extractor*
Three classifiers were trained, using supervised learning, based on the 4096 feature values that were extracted from each of AlexNet's two FC layers FC6 and FC7 separately. The classification performance was evaluated with TP, FN, FP, TN, classification accuracy, training time and classification time as shown in Table 2.

Using the features from FC6, the highest classification accuracy of 97.0% was obtained with an SVM with a quadratic kernel; while the lowest classification accuracy of 90.8% was obtained with LDA. Likewise, using the features extracted from FC7, the highest classification accuracy of 95.8% was obtained with an SVM and a quadratic kernel; while the lowest classification accuracy of 91.9% was obtained with LDA. Using the features extracted from FC6 provided better classification accuracy with the SVMs and the random forest than using the features from FC7; while with LDA, using the features extracted from FC7 provided better classification accuracy than using the features from FC6.

**Table 1 – Confusion matrix used for sugar beet and volunteer potato classification.**

| | | Predicted class | |
|---|---|---|---|
| | | Sugar beet (SB) | Volunteer potato (VP) |
| Actual class | Sugar beet (SB) | TP | FN |
| | Volunteer potato (VP) | FP | TN |

TP: true-positive, TN: true-negative, FP: false-positive, and FN: false-negative.

The smallest FN and FP values were 21 and 12, respectively, which were obtained using the features extracted from FC6 and the SVM with a quadratic kernel. The FN number of 21 indicates that in total 3.8% of sugar beet was classified as volunteer potato, and thus would be eliminated by the weed control robot. The FP of 12 indicates that 2.2% of volunteer potato was classified as sugar beet, and thus would not be killed.

The training time includes the time needed for feature extraction as well as training of the classifier itself. Using the features extracted from FC6, the SVMs and LDA required 13–14 s of training time while the random forest required 16 s of training time. Similarly, using the features extracted from FC7, the SVMs and LDA required 15 s of training time while the random forest required 17 s of training time. The average training time for one plant image was 0.014 s. The training times needed by all classifiers are reasonable, considering the training can be done offline and may not have to be repeated very often.

The classification time indicates the time required to classify (or predict) the class of a single plant image using a trained classifier. For all classifiers, an average of 0.016 s was needed using the features extracted from FC6, and an average of 0.018 s was needed using the features extracted from FC7. This classification time is fast enough for real-time application in the field (classification time < 0.1 s).

**Table 2 – Scenario 1: the classification performance is shown using features that were extracted from each of AlexNet's two FC layers in FC6 and FC7 separately. The classifiers were trained and validated with a total of 1100 images (550 of sugar beet and 550 of volunteer potato) using 10-fold random cross-validation. The final classification performance was averaged over ten repetitions. The training time includes times for feature extraction as well as training of the classifier. The classification time was measured for the time required to classify the class of a single plant image using a trained classifier.**

| Input layer and classifier models | | | TP | FN | FP | TN | Classification accuracy (%) | Training time (s) | Classification time (s/image) |
|---|---|---|---|---|---|---|---|---|---|
| | | | (% of total) | | | | | | |
| FC6 | SVM | Linear | 526 (95.6) | 24 (4.4) | 18 (3.3) | 532 (96.7) | 96.2 | 13.3 | 0.0143 |
| | | Quadratic | 529 (96.2) | 21 (3.8) | 12 (2.2) | 538 (97.8) | 97.0 | 13.3 | 0.0143 |
| | | Cubic | 527 (95.8) | 23 (4.2) | 16 (2.9) | 534 (97.1) | 96.5 | 13.3 | 0.0142 |
| | Random forest | | 513 (93.3) | 37 (6.7) | 45 (8.2) | 505 (91.8) | 92.5 | 15.5 | 0.0154 |
| | LDA | | 490 (89.1) | 60 (10.9) | 41 (7.5) | 509 (92.5) | 90.8 | 13.9 | 0.0217 |
| FC7 | SVM | Linear | 515 (93.6) | 35 (6.4) | 24 (4.4) | 526 (95.6) | 94.6 | 14.6 | 0.0160 |
| | | Quadratic | 523 (95.1) | 27 (4.9) | 19 (3.5) | 531 (96.5) | 95.8 | 14.6 | 0.0161 |
| | | Cubic | 524 (95.3) | 26 (4.7) | 21 (3.8) | 529 (96.2) | 95.7 | 14.6 | 0.0161 |
| | Random forest | | 512 (93.1) | 38 (6.9) | 47 (8.5) | 503 (91.5) | 92.3 | 16.7 | 0.0170 |
| | LDA | | 499 (90.7) | 51 (9.3) | 38 (6.9) | 512 (93.1) | 91.9 | 15.2 | 0.0229 |

TP: true-positive, TN: true-negative, FP: false-positive, and FN: false-negative.

### 6.1.2. *Scenario 2 – Modified and fine-tuned AlexNet as a binary classifier*

The classification performance of the modified and fine-tuned AlexNet is shown in Fig. 8. As expected, when the number of training images increased from 200 to 900, the classification accuracy increased from 89.1% to 98.0%. However, the classification accuracy did not linearly increase with the number of training images. The largest improvement in classification accuracy (4.4%) was found when the number of training images changed from 200 to 300; while the smallest improvement in classification accuracy (0.3%) was found when the number of training images was changed from 800 to 900.

The highest classification accuracy obtained in scenario 1 was 97.0% as shown in Table 2. However, in scenario 2, a classification accuracy higher than 97.0% was only obtained when more than 700 training images were used.

The training time required for fine-tuning of the AlexNet linearly increased with the number of training images. For fine-tuning with 200 and 900 images, a training time of 94.9 s and 656.4 s was needed, respectively. The average training time for one plant image was 0.6 s. Comparing this training time with the training time in scenario 1 (0.014 s), training the deep network was found to be computationally more expensive than training the conventional classifiers.

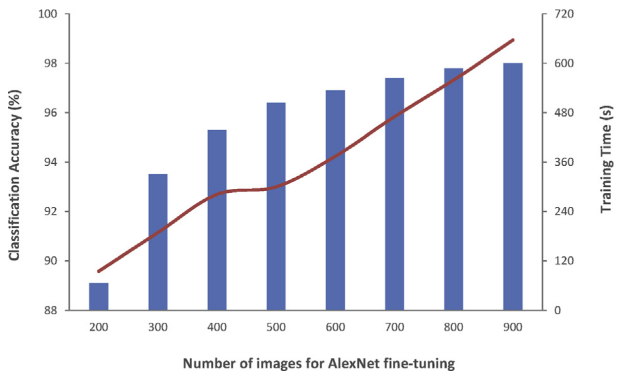

**Fig. 8 – Scenario 2 – AlexNet was modified to produce binary output for the classification of sugar beet and volunteer potato. The modified AlexNet was fine-tuned with a varying number of training images. The bars are classification accuracy and the line is training time as a function of the number of images for fine-tuning of AlexNet.**

In all cases, the classification time required to classify (or predict) the class of a single plant was 0.012 s, showing the fastest classification time among all scenarios.

### 6.1.3. *Scenario 3 – Modified and fine-tuned AlexNet as a fixed feature extractor*

The classification performance when the modified AlexNet was fine-tuned with 300 plant images is shown in Table 3. Using the features extracted from FC6, the highest classification accuracy of 96.7% was obtained with an SVM and linear kernel; while the lowest classification accuracy of 93.0% was obtained with the random forest and LDA. A similar trend in the results was found when using the features from FC7. The highest classification accuracy of 96.3% was obtained with SVM and linear kernel; while the lowest classification accuracy of 91.0% was obtained with LDA.

In Table 3, the smallest FN and FP were 6 and 4, respectively, which were obtained using the features extracted from FC6 and the SVM with a linear kernel. The FN value of 6 indicates that in total 4.0% of sugar beet was classified as volunteer potato, and thus would be killed by the weed control robot. The FP value of 4 indicates that 2.7% of volunteer potato was classified as sugar beet, and thus would be left untreated by the weed control robot.

In Table 4, the classification performance is shown when the modified AlexNet was fine-tuned with 800 plant images. Using the features extracted from FC6, the highest classification accuracy of 97.3% was obtained with the SVM and linear kernel; while the lowest classification accuracy of 95.3% was obtained with LDA. Using the features from FC7, the highest classification accuracy of 97.3% was obtained with random forest; while the lowest classification accuracy of 96.0% was obtained with the LDA.

In Table 4, the smallest FN and FP were 4 and 4, respectively, which were obtained using the features extracted from

**Table 3 – Scenario 3: After fine-tuning of the AlexNet with 300 images, the classifiers were trained with features extracted each from FC6 and FC7 separately. A total of 300 training images for fine-tuning was randomly selected from 1100 plant images in the dataset. From the remaining images, a total of 300 images was randomly selected for classifier training and validation. The training time includes times for fine-tuning of AlexNet and training of the classifier. The classification time was measured for the time required to classify the class of a single plant image using a trained classifier.**

| Layer and classifier models | | | TP | FN | FP | TN | Classification accuracy (%) | Training time (s) | Classification time (s/image) |
|---|---|---|---|---|---|---|---|---|---|
| | | | (% of total) | | | | | | |
| FC6 | SVM | Linear | 144 (96.0) | 6 (4.0) | 4 (2.7) | 146 (97.3) | 96.7 | 195.8 | 0.0130 |
| | | Quadratic | 142 (94.7) | 8 (5.3) | 5 (3.3) | 145 (96.7) | 95.7 | 196.3 | 0.0131 |
| | | Cubic | 142 (94.7) | 8 (5.3) | 6 (4.0) | 144 (96.0) | 95.3 | 195.2 | 0.0130 |
| | Random forest | | 140 (93.3) | 10 (6.7) | 11 (7.3) | 139 (92.7) | 93.0 | 198.5 | 0.0134 |
| | LDA | | 138 (92.0) | 12 (8.0) | 9 (6.0) | 141 (94.0) | 93.0 | 197.9 | 0.0180 |
| FC7 | SVM | Linear | 143 (95.3) | 7 (4.7) | 4 (2.7) | 146 (97.3) | 96.3 | 196.3 | 0.0144 |
| | | Quadratic | 143 (95.3) | 7 (4.7) | 5 (3.3) | 145 (96.7) | 96.0 | 197.1 | 0.0143 |
| | | Cubic | 142 (94.7) | 8 (5.3) | 6 (4.0) | 144 (96.0) | 95.3 | 196.4 | 0.0143 |
| | Random forest | | 143 (95.3) | 7 (4.7) | 9 (6.0) | 141 (94.0) | 94.7 | 197.9 | 0.0146 |
| | LDA | | 135 (90.0) | 15 (10.0) | 12 (8.0) | 138 (92.0) | 91.0 | 196.4 | 0.0183 |

TP: true-positive, TN: true-negative, FP: false-positive, and FN: false-negative.

**Table 4 – Scenario 3: After fine-tuning of the AlexNet with 800 images, the classifiers were trained with features extracted each from FC6 and FC7 separately. A total of 800 training images for fine-tuning was randomly selected from 1100 plant images in the dataset. The remaining images, 300 images, were used for classifier training and validation. The training time includes times for fine-tuning of AlexNet and training of the classifier. The classification time was measured for the time required to classify the class of a single plant image using a trained classifier.**

| Layer and classifier models | | | TP | FN | FP | TN | Classification accuracy (%) | Training time (s) | Classification time (s/image) |
|---|---|---|---|---|---|---|---|---|---|
| | | | (% of total) | | | | | | |
| FC6 | SVM | Linear | 145 (96.7) | 5 (3.3) | 3 (2.0) | 147 (98.0) | 97.3 | 581.4 | 0.0135 |
| | | Quadratic | 146 (97.3) | 4 (2.7) | 5 (3.3) | 145 (96.7) | 97.0 | 584.8 | 0.0135 |
| | | Cubic | 145 (96.7) | 5 (3.3) | 5 (3.3) | 145 (96.7) | 96.7 | 583.2 | 0.0136 |
| | Random forest | | 145 (96.7) | 5 (3.3) | 6 (4.0) | 144 (96.0) | 96.3 | 586.2 | 0.0140 |
| | LDA | | 142 (94.7) | 8 (5.3) | 6 (4.0) | 144 (96.0) | 95.3 | 584.9 | 0.0204 |
| FC7 | SVM | Linear | 145 (96.7) | 5 (3.3) | 4 (2.7) | 146 (97.3) | 97.0 | 583.9 | 0.0148 |
| | | Quadratic | 146 (97.3) | 4 (2.7) | 5 (3.3) | 145 (96.7) | 97.0 | 584.5 | 0.0148 |
| | | Cubic | 146 (97.3) | 4 (2.7) | 5 (3.3) | 145 (96.7) | 97.0 | 585.4 | 0.0149 |
| | Random forest | | 146 (97.3) | 4 (2.7) | 4 (2.7) | 146 (97.3) | 97.3 | 586.9 | 0.0159 |
| | LDA | | 143 (95.3) | 7 (4.7) | 5 (3.3) | 145 (96.7) | 96.0 | 585.8 | 0.0221 |

TP: true-positive, TN: true-negative, FP: false-positive, and FN: false-negative.

FC7 and using the random forest classifier. This misclassification indicates that 2.7% of sugar beet would be treated by the weed control device; while 2.7% of volunteer potato would not be treated. It should be noted that the same classification accuracy of 97.3% was achieved using FC6 with SVM linear and using FC7 with random forest. However, different FN and FP were obtained. Using FC6 with SVM linear, FN and FP values were 5 and 3, representing misclassification of 3.3% of sugar beet and 2.0% of volunteer potato, respectively. Furthermore, using FC7 with random forest, FN and FP values were 4 and 4, representing misclassification of 2.7% of sugar beet and 2.7% of volunteer potato, respectively.

Fine-tuning with 800 images produced better classification accuracy in all classifiers when compared to using 300 images for fine-tuning: a 0.6% and 1.3% increase in the highest classification accuracies using the features from FC6 and FC7, respectively; and a 2.4% and 4.7% increase in the lowest classification accuracies obtained when using the features from FC6 and FC7, respectively.

The training time includes the time needed for fine-tuning of AlexNet as well as training of the classifier. Using 300 images for fine-tuning, 195–197 s of training time was needed for all classifiers; while using 800 images for fine-tuning, 583–586 s of training time was needed for all classifiers. The average time for training the classifiers was 4.3 s showing that the major portion of the required training time was used for fine-tuning of AlexNet. Again, training the deep network is a computationally more expensive task than training conventional classifiers.

The classification time required to classify a single plant image was 0.013–0.022 s for all classifiers. SVMs showed the shortest classification time, while LDA showed the longest classification time among all classifiers. These classification times are fast enough for real-time application in the field (classification time < 0.1 s).

### 6.1.4. *All scenarios: summary*

The classification accuracies are summarised in Fig. 9. Among all scenarios, both the highest and lowest classification

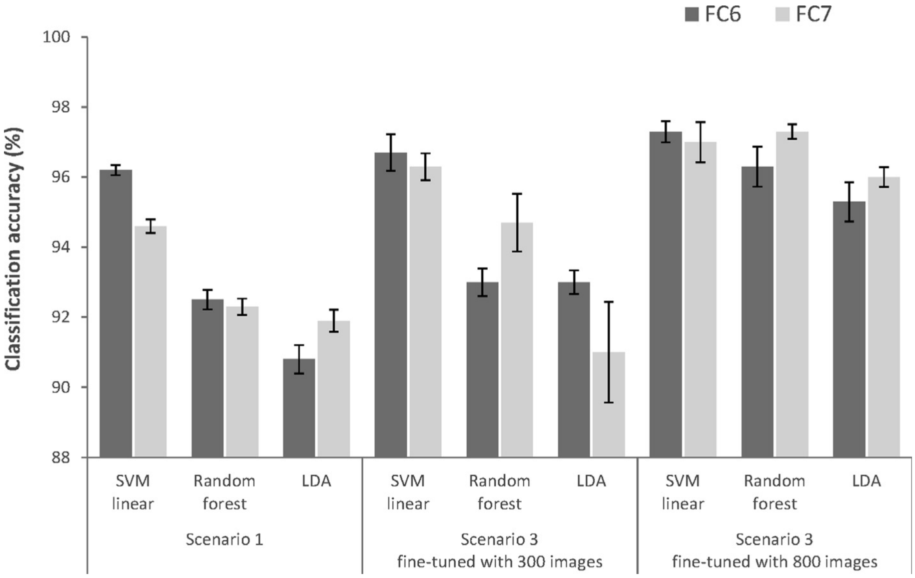

**Fig. 9 – The obtained classification accuracies using SVM with a linear kernel, random forest and LDA in scenario 1 and 3 are summarised. The classifiers were trained using the features extracted each from FC6 (FC layer 6) and FC7 (FC layer 7) separately. In scenario 3, the AlexNet was fine-tuned with 300 and 800 images separately.**

accuracies were obtained in scenario 2. The highest classification accuracy achieved in scenario 2 was 98.0% while highest accuracy in scenarios 1 and 3 were 97.0% and 97.3%, respectively. In scenario 2, a higher classification accuracy than 97.3% was obtained when the number of images used for fine-tuning was more than 700. On the other hand, the lowest classification accuracy achieved in scenario 2 was 89.1% while those in scenario 1 and 3 were 90.8% and 91.0%, respectively. When only a small training dataset was used, training the conventional classifier yielded better performance than fine-tuning the AlexNet. However, a large number of images for fine-tuning resulted in a better classification accuracy compared to the conventional classifier training.

Using the conventional classifiers in scenario 1 and 3, SVMs showed better classification accuracy than the other classifiers. However, the difference in classification accuracy among the classifiers tended to decrease as AlexNet was fine-tuned with more training images.

### 6.2. *Classification performance amongst different pre-trained networks*

The classification performance of the modified and fine-tuned deep networks (based on scenario 2) is shown in Table 5.

When the training was stopped after 20 epochs, the highest classification accuracy of 98.4% was obtained with VGG-19; while the lowest classification accuracy of 90.8% was obtained with Inception-v3. AlexNet showed a classification accuracy of 97.9% which was higher than the accuracy obtained with the same number of training data (500 images) in scenario 2 (96.4%). This result indicates that applying data augmentation may yield a better classification accuracy. Training time varied from 9 to 106 min with AlexNet requiring the shortest training time and ResNet-101 requiring the longest training time. These results for training time are reasonable since AlexNet contains the smallest number of layers, whereas ResNet-101 contains the largest number of

**Table 5 – The classification performance among six pre-trained deep networks was evaluated with two training epochs (20 and 30). Based on scenario 2, each network was modified and fine-tuned to classify sugar beet and volunteer potato. Randomly selected 500 images were used for training, while the remaining 600 images were used for validation. The classification performance was averaged over five repetitions and validated with classification accuracy, training time and classification time.**

| | Training 20 epoch | | | Training 30 epoch | | |
|---|---|---|---|---|---|---|
| | Accuracy (%) | Training time (min) | Classification time (s/image) | Accuracy (%) | Training time (min) | Classification time (s/image) |
| AlexNet | 97.9 | 9.0 | 0.0038 | 97.7 | 15.6 | 0.0040 |
| VGG-19 | 98.4 | 37.4 | 0.0130 | 98.7 | 71.4 | 0.0124 |
| GoogLeNet | 97.0 | 23.8 | 0.0033 | 97.3 | 36.9 | 0.0035 |
| ResNet-50 | 96.2 | 40.3 | 0.0072 | 97.2 | 69.8 | 0.0075 |
| ResNet-101 | 97.5 | 106.6 | 0.0118 | 98.5 | 162.0 | 0.0111 |
| Inception-v3 | 90.8 | 88.7 | 0.0088 | 94.8 | 133.0 | 0.0086 |

layers of all. Interestingly enough, GoogLeNet required less training time than VGG-19 even though GoogLeNet contained more layers than VGG-19. Also, in classification time, GoogLeNet needed far less time for classification than VGG-19. In fact, GoogLeNet required the shortest classification time, even less than AlexNet, while VGG-19 required the longest classification time among all networks.

When the training was stopped after 30 epochs, again the highest classification accuracy of 98.7% was obtained with VGG-19; while the lowest classification accuracy of 94.8% was obtained with Inception-v3. However, this accuracy obtained with Inception-v3 was greatly improved compared to when the training was stopped after 20 epochs. Yet AlexNet, VGG-19 and GoogLeNet did not yield such improvements.

The values in Table 5 indicate an average over five repetitions. Together with the stochastic nature of the training, this will result in some variation in the accuracy. This probably explains the decrease in accuracy with AlexNet between when the training was stopped after 20 epochs (97.9%) and 30 epochs (97.7%).

In Fig. 10, the loss and accuracy for each epoch of the training with Inception-v3 and VGG-19 are shown. The accuracy for Inception-v3 still gradually improved even after 20 epochs; while the accuracy for VGG-19 more or less stabilised after only a small number of epochs. Likewise, the loss for Inception-v3 slowly reduced even after 20 epochs; while the loss for VGG-19 rapidly reduced from the first to five epochs and then reasonably stabilised although values were fluctuating a bit between zero and 0.15. This result indicates that Inception-v3 requires more epochs to reach the highest accuracy and lowest loss than VGG-19. A similar trend in the results was also found with ResNet-50 and ResNet-101. Again, GoogLeNet required less training time than VGG-19. Also, in classification time, GoogLeNet still required the shortest classification time, even less than AlexNet, while VGG-19 required the longest classification time among all networks. The classification time for all networks was found to be fast enough for real-time application in the field.

## 7. Discussion

The classification performance obtained using transfer learning in this study exceeds previously reported accuracies, for instance by Persson and Åstrand (2008), Nieuwenhuizen et al. (2010), and Suh et al. (2018). Given the widely varying circumstances in natural fields, the highest classification accuracy (98.7%) obtained in this study is considerably better, to the best of our knowledge, than any other approaches mentioned in the literature for crop and weed classification. To further substantiate the claim of considerable progress being made by using transfer learning, it would be beneficial to compare different algorithms, including ones previously used, on the same dataset. However, associating previously mentioned algorithms with this study might not yield a proper comparison because most, if not almost all, algorithms developed so far have been based on image acquisition hardware including a hood covering the scene from ambient light and by illuminating the scene with artificial light. Algorithms were tuned for that purpose and for those specific conditions. In the current research approach, however, no hood or artificial lighting was used. To compare algorithms on the current dataset would require retraining of the previously used algorithms which would not result in a fair comparison of results. Though a proper comparison is lacking, it seems fair to claim that, with transfer learning of ConvNet, progress can be made in this field.

The proposed approach in this study was a partial implementation of a full pipeline for weed detection. The obtained results were based on manually extracted plant images, and vegetation segmentation procedure was not integrated. Implementing a full pipeline may potentially reduce the overall performance. In other words, the proposed approach does not lead to the precise detection of volunteer potato in field images. The individual plant detection procedure needs to be integrated as well. Sa et al. (2016) proposed a fruit detection system using ConvNet that detects each fruit in the large image even under occlusion. A similar approach could be used to detect each individual plant in crop fields.

### 7.1. *Proper scenario selection in transfer learning*

When using transfer learning with ConvNets for weed classification, the most appropriate scenario needs to be selected based on the number of available training images. The highest classification accuracy was obtained in scenario 2, but only if a large number of images were used for fine-tuning. If a large number of images is not available, scenarios 1 and 3 would

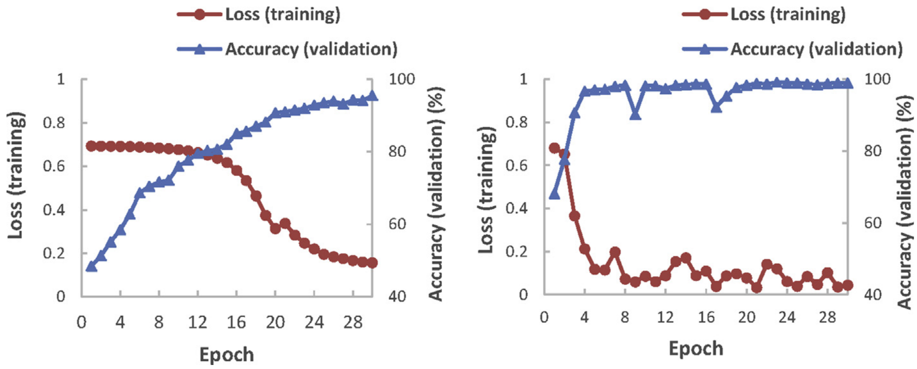

**Fig. 10 – The loss and accuracy for each epoch of the training with Inception-v3 (left) and VGG-19 (right).**

provide a better classification accuracy than scenario 2. Determining the required number of training images for the selection of scenarios may not be trivial, and perhaps may need more study as well. At least in this scenario testing, 700 training images were needed in scenario 2 to obtain better classification accuracy than in scenarios 1 and 3. However, by applying data augmentation as in the ConvNet comparison, even 500 training images were enough to obtain higher classification accuracy than in scenarios 1 and 3. Nevertheless, some studies have not applied data augmentation as the performance improvement was not considered to be significant (Wan, Zeiler, Zhang, LeCun, & Fergus, 2013; Wang, Luo, Huang, Zhao, & Wang, 2017; Yosinski et al., 2014).

In addition, the number of classes (plant species in this case) may need to be considered for a proper scenario selection as well. Hasan et al. (2016) reported that when the number of classes was large, the performance of ConvNet was worse than SVM classifier in their classification task. It is unclear how a different number of classes would influence the performance in crop and weed classification. Only binary classification was performed in this study based on the assumption that in most cases plants found in sugar beet fields, especially in The Netherlands, are either sugar beet or volunteer potato. However, in some agricultural fields, several different weed species are often found together. A future study topic might be the multiclass classification for crop and several weed species as well as the classification performance assessment among a different number of plant species.

Using an SVM in scenarios 1 and 3, the extracted features from FC6 provided better classification accuracy than using features from FC7. Hu et al. (2015) also reported that the extracted features from the first FC layer consistently provided better performance compared to the second FC layer. However, when using other classifiers such as random forest and LDA, the extracted features from FC7 provided better classification accuracy than the ones from FC6. The behaviour or function of each layer in the deep network is not yet fully understood, and the deep network is still seen as a "black-box." Research has revealed that the first layer in many deep neural networks, when trained on images, tends to learn general features similar to the Gabor filter and colour blobs (Yosinski et al., 2014). More understanding of the function of each layer is, therefore, a topic for future study.

### 7.2. *Different ConvNets architectures for weed classification*

AlexNet showed a classification accuracy of 97.9% in the ConvNet comparison. Considering the fact that AlexNet contains far fewer layers than the other networks used in the study, the classification accuracy obtained with AlexNet was surprisingly good even compared to the top performers such as VGG-19 (98.7%) and ResNet-101 (98.5%). Moreover, the training time required by AlexNet was considerably less than for the others. This training time can even be further reduced, without sacrificing the performance, if training stops after only 5–10 epochs since the accuracy during training was shown to be more or less stabilised after 5 epochs. Regarding the classification time, AlexNet was one of the fastest networks for classification, followed by GoogLeNet but only by a very small margin, which suits real-time applications very well. Given these results, it seems fair to use AlexNet in our application for the classification of sugar beet and volunteer potato, although AlexNet is already considered to be an "old-fashioned" network.

The number of epochs for training largely influences the classification performance depending on the network architecture. To reach the highest desired accuracy, some shallow networks such as AlexNet and VGG-19 require only a small number of epochs; while some deeper networks such as ResNet-101 and Inception-v3 require a relatively large number of epochs. It is unclear how to choose the optimum number of epochs for the training of deep networks since selecting the optimum number has been mainly based on empirical experience (Jozefowicz, Zaremba, & Sutskever, 2015; Schmidhuber, 2015). For this reason, monitoring the training process with the loss and accuracy is particularly important to determine when to stop the training. Also, training of a deep network depends on other parameter settings such as learning rate, momentum and batch size, which in many cases also relies on empirical knowledge (LeCun et al., 2015). All these parameter settings are likely to influence the classification performance which may also influence the required number of training images and epochs needed to obtain high classification accuracy. It is worth investigating in order to better understand the influence of various parameters used in the deep learning on the performance of the deep neural network.

VGG-19 is known to be an expensive and complex architecture regarding computational cost and number of parameters, which makes the network less suitable for real-time applications (Canziani, Paszke, & Culurciello, 2016). He et al. (2016a) discussed the fact that VGG-19 has higher complexity and requires more computations than ResNet-101, even though VGG-19 has considerably fewer layers than ResNet. This was confirmed in our ConvNet comparison (Table 5) as VGG-19 needed the longest classification time compared to the other networks.

According to Yosinski et al. (2014), the effectiveness of transfer learning is expected to decline if there is less similarity between the network's original task and the new task in hand. All networks in our study were originally (pre-)trained with ImageNet Dataset which contained object images commonly found in ordinary life such as desk, computer, animals, etc. The ImageNet Dataset is quite distinctly different, so to speak, from sugar beet and volunteer potato images; yet, the performance obtained using transfer learning in this study is still very impressive. If the networks were (pre-)trained with a crop/weed field image dataset, further promising performance may be achieved as similarity will be greater between the network's original task and the new task in hand.

### 7.3. *Practical considerations for weed control*

For weed control in practice, it is critical to have as large as possible a number of TPs as well as large as possible a number of TNs. Not only that, but it is also important to consider both the number of FNs (the number of sugar beet plants that are classified as volunteer potatoes) and the number of FPs (the number of volunteer potato plants that are classified as sugar beet). The FNs lead to the removal of the cash crop caused by the misclassification, thus keeping the number of FNs as small as possible is critical (Lottes et al., 2016). At the same time, however, maintaining the number of FPs as small as possible is also desired. If there are many leftover volunteer potato plants caused by misclassification, the weed control robot may have to drive repeatedly through the field in order for Dutch farmers to meet the statutory regulation in the Netherlands (Nieuwenhuizen, 2009). The economic consequences of the different numbers of FNs and FPs deserve further research.

Training and application of a deep neural network require sophisticated hardware; high-performance GPUs. This requirement has been a limiting factor in many applications (Sa et al., 2016). However, cloud services (e.g. Amazon Elastic Compute Cloud and Paperspace GPU Cloud), as used in this study, provide a simple and easy way of using high-performance computing hardware without having to acquire and maintain the hardware on site.

Although the calculation time was measured on cloud servers in this study, it is reasonable to think that a similar calculation speed could be achieved during in-field application because high-performance PCs (e.g. gaming laptops with high-performance GPUs) are available in the market that compare well to the cloud servers used in this study.

## 8. Conclusion

This study evaluated a transfer learning procedure and assessed the performance amongst different ConvNet architectures for the classification of sugar beet and volunteer potato under ambient varying light conditions. Three different implementation scenarios were assessed using AlexNet, and the performance of the following six pre-trained networks was compared: AlexNet, VGG-19, GoogLeNet, ResNet-50, ResNet-101 and Inception-v3.

Transfer learning provided very promising performance for the classification of sugar beet and volunteer potato images under ambient varying light conditions. In the scenario comparison, the highest classification accuracy (98.0%) was obtained with AlexNet in Scenario 2. In scenarios 1 and 3, the highest classification accuracies were 97.0% and 97.3%, respectively.

All three scenarios were feasible for real-time field applications (the classification time < 0.1 s), but training the deep network was a computationally more expensive task than training the conventional classifiers.

The highest classification accuracy (98.7%) obtained in the ConvNet comparison was, to the best of our knowledge, considerably better than any other approaches mentioned in the literature for crop and weed classification. Data augmentation may improve the classification accuracy. VGG-19 yielded the highest classification accuracy but needed the longest classification time. AlexNet required the shortest training time, while ResNet-101 required the longest training time. With Inception-v3 using 30 epochs instead of 20 epochs for training yielded a significant improvement in performance. Such improvements were not observed when using more training epochs with AlexNet, VGG-19 and GoogLeNet.

Three different scenarios as well as six different ConvNet architectures for transfer learning showed robust performance with the plant images acquired in different periods of the various years with two types of soils. However, implementing a full pipeline for weed detection may potentially reduce the overall performance.

## Acknowledgements

The work presented in this paper was part of the Agrobot part of the Smartbot project and funded by Interreg IVa, European Fund for the Regional Development of the European Union and Product Board for Arable Farming. We thank Gerard Derks at experimental farm Unifarm of Wageningen University for arranging and managing the experimental fields.

## REFERENCES

Ahmed, F., Al-Mamun, H. A., Bari, A. S. M. H., Hossain, E., & Kwan, P. (2012). Classification of crops and weeds from digital images: A support vector machine approach. *Crop Protection, 40*, 98–104. https://doi.org/10.1016/j.cropro.2012.04.024.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Hasan, M., Van Esesn, B. C., et al. (2018). *The history began from AlexNet: A comprehensive survey on deep learning approaches*. ArXiv. Retrieved from http://arxiv.org/abs/1803.01164.

Åstrand, B., & Baerveldt, A. J. (2002). An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots, 13*(1), 21–35. https://doi.org/10.1023/A:1015674004201.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding, 110*(3), 346–359. https://doi.org/10.1016/j.cviu.2007.09.014.

Canziani, A., Paszke, A., & Culurciello, E. (2016). *An analysis of deep neural network models for practical applications*. ArXiv. Retrieved from http://arxiv.org/abs/1605.07678.

Chen, L., Wu, C., Fan, W., Sun, J., & Naoi, S. (2014). Adaptive local receptive field convolutional neural networks for handwritten Chinese character recognition. *Pattern Recognit Commun Comput Inf Sci, 484*, 455–463. https://doi.org/10.1007/978-3-662-45643-9_48.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., et al. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition. In *31st international conference on machine learning, ICML 2014* (pp. 647–655). Beijing, China: International Machine Learning Society (IMLS). Retrieved from http://arxiv.org/abs/1310.1531.

Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering, 151*, 72–80. https://doi.org/10.1016/j.biosystemseng.2016.08.024.

Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors (Switzerland), 17*(9). https://doi.org/10.3390/s17092022.

Gebhardt, S., & Kühbauch, W. (2007). A new algorithm for automatic Rumex obtusifolius detection in digital images using colour and texture features and the influence of image resolution. *Precision Agriculture, 8*, 1–13. https://doi.org/10.1007/s11119-006-9024-7.

Gong, Y., Jia, Y., Leung, T., Toshev, A., & Ioffe, S. (2014). Deep convolutional ranking for multilabel image annotation. In

*Proceedings of the international conference on learning representations (ICLR 2014)*. Banff, Canada: ICLR. Retrieved from http://arxiv.org/abs/1312.4894.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, Massachusetts: The MIT Press. https://doi.org/10.1007/s13218-012-0198-z (Chapter 15.2).

Grinblat, G. L., Uzal, L. C., Larese, M. G., & Granitto, P. M. (2016). Deep learning for plant identification using vein morphological patterns. *Computers and Electronics in Agriculture, 127*, 418–424. https://doi.org/10.1016/j.compag.2016.07.003.

Hasan, M., Kotov, A., Idalski Carcone, A., Dong, M., Naar, S., & Brogan Hartlieb, K. (2016). A study of the effectiveness of machine learning methods for classification of clinical interview fragments into a large number of categories. *Journal of Biomedical Informatics, 62*, 21–31. https://doi.org/10.1016/j.jbi.2016.05.004.

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. In *The 29th IEEE conference on computer vision and pattern recognition (CVPR 2016)* (pp. 770–778). Las Vegas, Nevada, USA: IEEE Computer Society. https://doi.org/10.1007/s11042-017-4440-4.

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Identity mappings in deep residual networks. In *European conference on computer vision (ECCV 2016)* (pp. 630–645). Amsterdam, The Netherlands: Springer International Publishing. https://doi.org/10.1007/978-3-319-46493-0_38.

Hu, F., Xia, G.-S., Hu, J., & Zhang, L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing, 7*(11), 14680–14707. https://doi.org/10.3390/rs71114680.

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science, 353*(6301). https://doi.org/10.1126/science.aaf7894.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd international conference on machine learning (ICML 2015)* (Vol. 37, pp. 2342–2350). Lille, France: JMLR: W&CP. https://doi.org/10.1109/CVPR.2015.7298761.

Kazmi, W., Garcia-Ruiz, F., Nielsen, J., Rasmussen, J., & Andersen, H. J. (2015). Exploiting affine invariant regions and leaf edge shapes for weed detection. *Computers and Electronics in Agriculture, 118*, 290–299. https://doi.org/10.1016/j.compag.2015.08.023.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105). Lake Tahoe, NV: Curran Associates, Inc. https://doi.org/10.1016/j.protcy.2014.09.007.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791.

Longchamps, L., Panneton, B., Samson, G., Leroux, G. D., & Thériault, R. (2009). Discrimination of corn, grasses and dicot weeds by their UV-induced fluorescence spectral signature. *Precision Agriculture, 11*(2), 181–197. https://doi.org/10.1007/s11119-009-9126-0.

Lottes, P., Hoeferlin, M., Sander, S., Muter, M., Schulze, P., & Stachniss, L. C. (2016). An effective classification system for separating sugar beets and weeds for precision farming applications. In *Proceedings – IEEE international Conference on Robotics and automation (ICRA 2016)* (Vol. 2016–June, pp. 5157–5163). Stockholm, Sweden: IEEE. https://doi.org/10.1109/ICRA.2016.7487720.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110. https://doi.org/10.1023/B: VISI.0000029664.99615.94.

Nieuwenhuizen, A. T. (2009). *Automated detection and control of volunteer potato plants*. PhD thesis. The Netherlands: Wageningen University.

Nieuwenhuizen, A. T., Hofstee, J. W., & Van Henten, E. J. (2010). Performance evaluation of an automated detection and control system for volunteer potatoes in sugar beet fields. *Biosystems Engineering, 107*(1), 46–53. https://doi.org/10.1016/j.biosystemseng.2010.06.011.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE conference on computer vision and pattern recognition (CVPR 2014)* (pp. 1717–1724). Columbus, OH: IEEE Computer Society. Retrieved from http://hal.inria.fr/hal-00911179/.

O'Keeffe, M. G. (1980). The control of Agropyron repens and broad-leaved weeds pre-harvest of wheat and barley with the isopropylamine salt of glyphosate. In *Proceedings 1980 British Crop Protection Conference – Weeds* (Vol. 15, pp. 53–60).

Papadomanolaki, M., Vakalopoulou, M., Zagoruyko, S., & Karantzalos, K. (2016). Benchmarking deep learning frameworks for the classification of high resolution satellite multispectral data. In *ISPRS annals of photogrammetry, remote sensing and spatial information sciences* (Vol. III, pp. 83–88). Prague, Czech Republic: ISPRS Congress. https://doi.org/10.5194/isprsannals-III-7-83-2016.

Pérez, A. J., López, F., Benlloch, J. V., & Christensen, S. (2000). Colour and shape analysis techniques for weed detection in cereal fields. *Computers and Electronics in Agriculture, 25*(3), 197–212. https://doi.org/10.1016/S0168-1699(99)00068-X.

Persson, M., & Åstrand, B. (2008). Classification of crops and weeds extracted by active shape models. *Biosystems Engineering, 100*(4), 484–497. https://doi.org/10.1016/j.biosystemseng.2008.05.003.

Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE computer society conference on computer vision and pattern recognition workshops (CVPR workshop 2014)* (pp. 512–519). Columbus, OH: IEEE Computer Society. https://doi.org/10.1109/CVPRW.2014.131.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). DeepFruits: A fruit detection system using deep neural networks. *Sensors*, 16(8), 1222. https://doi.org/10.3390/s16081222.

Sainath, T. N., Mohamed, A. R., Kingsbury, B., & Ramabhadran, B. (2013). Deep convolutional neural networks for LVCSR. In *Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP 2013)* (pp. 8614–8618). Vancouver, Canada: IEEE Signal Processing Society. https://doi.org/10.1109/ICASSP.2013.6639347.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks, 61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

Schwing, A. G., & Urtasun, R. (2015). *Fully connected deep structured networks*. Preprint arXiv:1503.02351v1. Retrieved from http://arxiv.org/abs/1503.02351.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). *OverFeat: Integrated recognition, localization and detection using convolutional networks*. Preprint arXiv:1312.6229. Retrieved from http://arxiv.org/abs/1312.6229.

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., et al. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging, 35*(5), 1285–1298. https://doi.org/10.1109/TMI.2016.2528162.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International conference on learning representations (ICLR 2015)* (pp. 1–14). San Diego, USA: ICLR. https://doi.org/10.1016/j.infsof.2008.09.005.

Slaughter, D. C. C., Giles, D. K. K., & Downey, D. (2008). Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture, 61*(1), 63–78. https://doi.org/10.1016/j.compag.2007.05.008.

Suh, H. K., Hofstee, J. W., IJsselmuiden, J., & Van Henten, E. J. (2016). Discrimination between volunteer potato and sugar beet with a bag-of-visual-words model. In *International conference on agricultural engineering, CIGR-AgEng 2016*. Aarhus, Denmark: International Commission of Agricultural and Biosystems Engineering.

Suh, H. K., Hofstee, J. W., IJsselmuiden, J., & Van Henten, E. J. (2018). Sugar beet and volunteer potato classification using bag-of-visual-words model, scale-invariant feature transform, or speeded up robust feature descriptors and crop row information. *Biosystems Engineering, 166*, 210–226. https://doi.org/10.1016/j.biosystemseng.2017.11.015.

Sun, Y., Liu, Y., Wang, G., & Zhang, H. (2017). Deep learning for plant identification in natural environment. *Computational Intelligence and Neuroscience, 2017*, 1–6. https://doi.org/10.1155/2017/7361042 (Article ID-7361042).

Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR 2014)* (pp. 1891–1898). Columbus, OH: IEEE Computer Society. https://doi.org/10.1109/CVPR.2014.244.

Swain, K. C., Nørremark, M., Jørgensen, R. N., Midtiby, H. S., & Green, O. (2011). Weed identification using an automated active shape matching (AASM) technique. *Biosystems Engineering, 110*(4), 450–457. https://doi.org/10.1016/j.biosystemseng.2011.09.011.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. In *The 31st AAAI conference on artificial intelligence (AAAI-17)* (pp. 4278–4284). San Francisco, USA: AAAI-17. https://doi.org/10.1016/j.patrec.2014.01.008.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *IEEE conference on computer vision and pattern recognition (CVPR 2015)* (pp. 1–9). Boston, MA: IEEE Computer Society. https://doi.org/10.1109/CVPR.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision. In *IEEE conference on computer vision and pattern recognition (CVPR 2016)* (pp. 2818–2826). Las Vegas, Nevada, USA: IEEE. https://doi.org/10.1109/CVPR.2016.308.

Vedaldi, A., & Lenc, K. (2015). MatConvNet: Convolutional neural networks for MATLAB. In *Proceedings of the 23rd annual ACM international conference on multimedia* (pp. 689–692). Brisbane, Australia: ACM Press. https://doi.org/10.1145/2733373.2807412.

Wang, J., Luo, C., Huang, H., Zhao, H., & Wang, S. (2017). Transferring pre-trained deep CNNs for remote scene classification with general features learned from linear PCA network. *Remote Sensing, 9*(3), 225. https://doi.org/10.3390/rs9030225.

Wan, L., Zeiler, M., Zhang, S., LeCun, Y., & Fergus, R. (2013). Regularization of neural networks using dropconnect. In *30th international conference on machine learning (ICML 2013)* (pp. 109–111). Atlanta, USA: JMLR: W&CP. https://doi.org/10.1109/TPAMI.2017.2703082.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data, 3*(1), 1–40. https://doi.org/10.1186/s40537-016-0043-6.

Wilf, P., Zhang, S., Chikkerur, S., Little, S. A., Wing, S. L., & Serre, T. (2016). Computer vision cracks the leaf code. *Proceedings of the National Academy of Sciences of the United States of America, 113*(12), 3305–3310. https://doi.org/10.1073/pnas.1524473113.

Xie, M., Jean, N., Burke, M., Lobell, D., & Ermon, S. (2016). Transfer learning from deep features for remote sensing and poverty mapping. In *30th AAAI conference on artificial intelligence* (pp. 3929–3935). Phoenix, Arizona: AAAI-16. Retrieved from http://arxiv.org/abs/1510.00098.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. In *Advances in neural information processing systems (NIPS 2014)* (pp. 3320–3328). Montreal, Canada: NIPS. Retrieved from http://arxiv.org/abs/1411.1792.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision (ECCV 2014)* (pp. 818–833). Zurich, Switzerland: Springer. https://doi.org/10.1007/978-3-319-10590-1_53.

Zhang, Z., Kodagoda, S., Ruiz, D., Katupitiya, J., & Dissanayake, G. (2010). Classification of Bidens in wheat farms. *International Journal of Computer Applications in Technology, 39*, 123–129. https://doi.org/10.1109/MMVIP.2008.4749584.