Computers and Electronics in Agriculture 171 (2020) 105306

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Towards weeds identification assistance through transfer learning

Borja Espejo-Garcia[a,*], Nikos Mylonas[a], Loukas Athanasakos[a], Spyros Fountas[a], Ioannis Vasilakoglou[b]

[a] Agricultural University of Athens, Athens, Greece
[b] Institute of Thessaly, Thessaly, Greece

ARTICLE INFO

Keywords:
Weed identification
Deep learning
Transfer learning
Open data
Precision agriculture

ABSTRACT

Reducing the use of pesticides through selective spraying is an important component towards a more sustainable computer-assisted agriculture. Weed identification at early growth stage contributes to reduced herbicide rates. However, while computer vision alongside deep learning have overcome the performance of approaches that use hand-crafted features, there are still some open challenges in the development of a reliable automatic plant identification system. These type of systems have to take into account different sources of variability, such as growth stages and soil conditions, with the added constraint of the limited size of usual datasets. This study proposes a novel crop/weed identification system that relies on a combination of fine-tuning pre-trained convolutional networks (Xception, Inception-Resnet, VGNets, Mobilenet and Densenet) with the "traditional" machine learning classifiers (Support Vector Machines, XGBoost and Logistic Regression) trained with the previously deep extracted features. The aim of this approach was to avoid overfitting and to obtain a robust and consistent performance. To evaluate this approach, an open access dataset of two crop [tomato (*Solanum lycopersicum* L.) and cotton (*Gossypium hirsutum* L.)] and two weed species [black nightshade (*Solanum nigrum* L.) and velvetleaf (*Abutilon theophrasti* Medik.)] was generated. The pictures were taken by different production sites across Greece under natural variable light conditions from RGB cameras. The results revealed that a combination of fine-tuned Densenet and Support Vector Machine achieved a micro F_1 score of 99.29% with a very low performance difference between train and test sets. Other evaluated approaches also obtained repeatedly more than 95% F_1 score. Additionally, our results analysis provides some heuristics for designing transfer-learning based systems to avoid overfitting without decreasing performance.

1. Introduction

Since the world population is growing continuously, the production of high nutritional food must be increased in the next years (Kitzes et al., 2008; United Nations (UN), 2019), accompanied with the protection of natural ecosystems by using sustainable farming procedures (Food and Agriculture Organization, 2013). Among these actions, one of the most challenging is an effective reduction of the impact caused by weeds. In recent years, their damage, alongside with other pests, accounts for about 40% of global yield losses and is expected to increase in the coming years (European Crop Protection Association (ECPA), 2017). Weeds compete with crop plants for resources such as water, nutrients, light and space, causing crop yield losses. Additionally, weeds reduce the quality of farm products, cause irrigation water loss and hinder the operation of harvesting machines and, consequently, decrease the commercial value of cultivated areas (Rizzardi and Fleck, 2004; Zimdahl, 2018). To deal with weeds, farmers tend to apply uniform herbicide spraying throughout the field, usually two or three times in each growing season. However, this practice has as a result to apply uncontrolled large quantities of herbicides, which is harmful for humans, non-target organisms and the environment (Oerke, 2006; Zimdahl, 2018).

Currently, a key objective is the deployment of new information and communication technologies solutions to reduce the reliance on agrochemical products and to reduce human errors caused by cognitive phenomena (Bock et al., 2010). Specifically, the combination of smartphones with high performance processors, HD cameras, sensors, and cloud computing capabilities allow weed detection, when suitable human assessment is unavailable. Consequently, a most effective weed management is possible, contributing to the improvement of farm productivity, food security and environmental sustainability (Gebbers and Adamchuk, 2010).

Various studies have been carried out aiming on an accurate, reproducible and automatic identification method for weeds. The

* Corresponding author.
E-mail address: borjaeg@aua.gr (B. Espejo-Garcia).

https://doi.org/10.1016/j.compag.2020.105306
Received 8 October 2019; Received in revised form 24 January 2020; Accepted 23 February 2020
Available online 06 March 2020

majority of these studies were mostly concerned with acquisition, preprocessing, extraction of manually-designed features and supervised classifiers (Haug et al., 2014; Lottes et al., 2016; Kounalakis et al., 2016; Kounalakis et al., 2017; Lottes et al., 2017; Bakhshipour and Jafari, 2018). Classifiers used for this purpose as Support Vector Machines (SVM) and traditional/shallow neural networks are dependent on good feature extractors, such as Scale-Invariant Feature Transform (SIFT). However, the main limiting factor for these weed recognition systems is the limited ability of their manually-designed extracted features (colour information, shape analysis or texture analysis) to effectively represent image content. This factor adds up to a variety of challenges associated with image classification, such as viewpoint, scale and intra-class variations, image deformation, image occlusion, illumination conditions, and background clutter.

Furthermore, most previous studies based on hand-crafted features require tedious operations in the case of extensive preprocessing, image normalization and plant segmentation, which may significantly restrict the repeatability of the identification method. In recent years, researchers on weed or other plant identification developed systems incorporating deep learning methodologies. According to Ferentinos, (2018), deep learning refers to "the use of artificial neural network, architectures that contain a quite large cascade of multiple processing layers, as opposed to shallower architectures of more traditional neural network methodologies". With the inception of deep learning concepts, the solution to the limitations of the hand-crafted approaches seems to be closer than ever, as a result of the automatic feature extraction from the raw data (Lecun et al., 2015). Among the deep learning tools for weed identification, the most commonly used are the Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012). This kind of neural network is complex but efficient, with a high rate of discrimination and have proven to provide good results in precision agriculture for the correct identification of plants. Compared to previous methods, the self-learned features make the CNN less affected by natural variations such as changes in illumination, skewed leaves and occluded plants. Currently, CNNs are used in end-to-end crop/weed identification systems to overcome the limitations of hand-crafted approaches and reaching state-of-the-art performance. Potena et al., (2016) used a cascade of CNNs for crop/weed identification, where the first CNN detected vegetation and then the vegetation pixels were classified by a deeper crop-weed CNN. They used a multispectral camera to obtain RGB and NIR images. McCool et al., (2017) fine-tuned a very deep CNN and gained practical processing times by compression of the fine-tuned network using a mixture of small, but fast networks, without losing too much identification accuracy. Fully CNNs directly estimate a pixel-wise segmentation of the complete image and can use information from the whole image. Finally, Milioto et al., (2018) proposed a deep encoder-decoder CNN that exploits existing vegetation indexes and provides an identification in real-time.

Another important breakthrough has been the use of transfer learning in order to relax the data requirement of the CNNs (Ferentinos, 2018). Transfer learning recycles previously trained networks by using the new data to update a small part of the original weights (Bengio, 2012). The new features learned can prove useful for many different problems, even though these new problems may involve completely different classes than those of the original task. Wang et al., (2017) used transfer learning in order to obtain the best neural-based method for disease detection in plants. Our work is highly related with Suh et al., (2018), where the authors deeply compared different transfer learning approaches in order to find a suitable approach for weed detection (volunteer potato). Finally, another relevant study has been performed by Kounalakis et al., (2019) where they evaluated transfer learning by a combination of CNN-based feature extraction and linear classifiers to recognize rumex under real-world conditions.

In this paper, we present a novel methodology to find a suitable neural network that automatically performs the crop/weed identification tasks in real-time. To face this fine-grained image classification, we used the combination of fine-tuned deep neural networks for feature extraction with other machine learning classifiers on the top of them. To explore the best architecture and training mechanism, we fine-tuned some of the state-of-the-art deep neural networks. Our approach exploits a pipeline that includes different CNNs applied to the input images, where we can find tomato (*Solanum lycopersicum* L.), cotton (*Gossypium hirsutum* L.) and two weeds in relation to them: the black nightshade (*Solanum nigrum* L.) and velvetleaf (*Abutilon theophrasti* Medik.). Additionally, since there are few public datasets for weeds and multi-spectral cameras are still too expensive for small farmers, we have developed our own dataset to perform the experiments and validate our methodology. We have taken data from different farms in order to confirm the effectiveness of our approach. The pictures were obtained under natural variable light conditions from an RGB camera, which is the main difference from other datasets which use NIR or RGB + NIR cameras (Potena et al., 2016; Haug and Ostermann, 2018). However, it is important to underpin that the approach presented in this paper is also applicable for multi-spectral images.

In summary, our contribution is three-fold. Firstly, we have developed a novel vision-based solution for automated crop/weed identification from unconstrained RGB plant photographs by utilising fine-tuned deep neural networks. Secondly, we made an in-depth performance evaluation of the critical factors affecting the fine-tuning of pre-trained models, such as background removal, pooling type, and strategy for transfer learning. Our findings determine the relative significance of each of the aforementioned variables on performance, thus paving the way for more efficient utilisation of valuable computational resources. Finally, we provide an open dataset with crop/weed RGB images under natural variable light conditions where our methodology and future research could be evaluated.

This article is structured as follows: Section 2 presents our transfer learning approach and the crop/weed dataset. Section 3 explains the experimental setup and results. Section 4 presents a discussion about the results and in Section 5 conclusions are drawn and future directions envisaged.

2. Material and methods

The primary goal of this research was to develop a combination of pre-trained deep neural networks and a classifier to robustly identify crops and weeds in the field, even when the visual appearance of the crops and soil has changed. As shown in Fig. 1, the approach is composed of different stages. At each stage there are different configurations that were empirically evaluated to find the best crop/weed classifier. The first stage consisted of the capture of crops and weeds images, and their annotation in order to create the dataset. The second stage consisted of selecting the best segmentation approach for the image. This step aims to ease the next step of feature extraction. In the third stage, different pre-trained neural networks were fine-tuned in order to obtain the best feature extraction method. The fourth stage consisted of the training of machine learning classifiers that use the deep features extracted by the previously fine-tuned neural networks. Finally, the last stage consisted of the deployment of the whole weed detection system. The implementation of this last stage was out of the scope of this paper, but will be discussed later, taking into account different possibilities as well as the smartphones' constraints to achieve a real-time assistance application.

2.1. Dataset description

The data acquisition was carried out at 3 different farms in Greece. In Table 1, there is a specification of the geographical coordinates of the locations. Pictures were taken weekly in May and June 2019, at 8-to 10 a.m., to ensure similar light intensity in every session. The experiment was conducted on different fields. There was a sufficient geographic coverage of the experiment, as the 3 main production areas in Greece

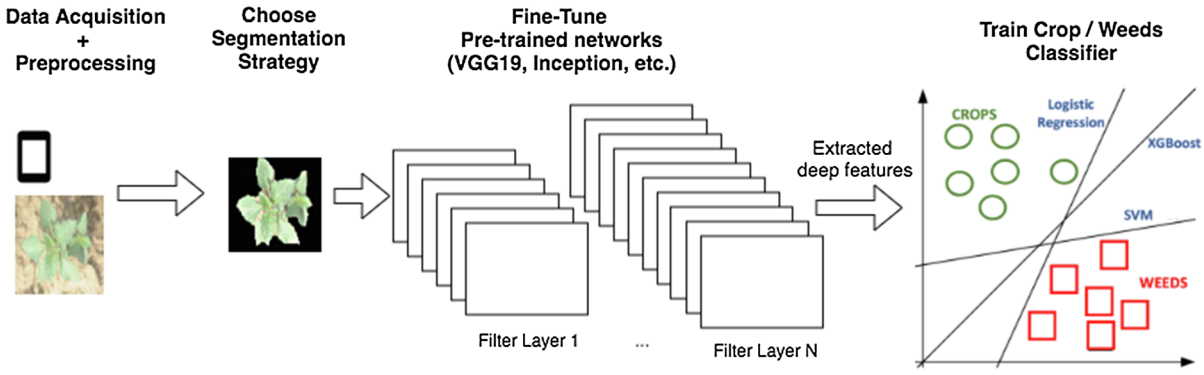

Fig. 1. Our proposed pipeline to find the most suitable combination of feature extractor and classifier.

Table 1
Location of experimental fields.

Region	Lat/Lon
South-central Greece	38°21′50.1″N 23°20′26.4″E
Central Greece	39°36′05.6″N 22°34′04.7″E
Northern Greece	40°36′02.5″N 22°18′20.0″E

Table 2
Description of image acquisition and main dataset statistics.

Parameter	Value
No. Tomato/Cotton/Wild Tomato/Wild Cotton	202/48/130/124
Camera Models	Nikon D700 (2272X1704)
Weather Conditions	Sunny
Start Time	15.03/16.05
Avg. RGB Channels	(179.80, 166.69, 138.39)

have been considered. The value crops in these fields were the tomato and cotton, while the target weeds were the black nightshade and velvetleaf, which pose very competitive weeds for the aforementioned crops.

Due to the importance of weed identification at early growth stage, the weeds were pictured at the 3 to 4 leaves stage. The pictures had been taken with a Nikon D700 camera that delivers 12 mega pixel images. The images were taken from approximately 1 m height and they differ from each other in terms of soil type. Moreover, in all image acquisition sessions the light conditions are characterized as clear, without overcast or illumination variations due to clouds. One image example of each class from the dataset are presented in Fig. 2. This dataset is rather variant because different photographers have collected images from different locations under varying conditions of soil, colour, and illumination. A detailed description of the data and the setup for its acquisition, is provided in Table 2. We have published the dataset[1] to contribute to the research community with valuable data and expand the research approaches.

2.2. Plant segmentation

After data acquisition and dataset creation, the next step in the methodology was to segment the crop/weed by removing the background. Although theoretically this ability should be executed by the neural network, the reality is that some studies have shown that the network could learn to recognise the background, which will cause the undesirable bias towards some classes, when the network detects specific backgrounds (Mohanty et al., 2016; Barbedo, 2018). In this work, we contrasted empirically the use of plant segmentation against the use of the original image without any preprocessing step.

Specifically, we used the method illustrated in Fig. 3 to discern between vegetation and ground pixels, and, as a consequence, segment the plant. Initially, the R (Red), G (Green) and B (Blue) channel values in the images are normalized using (Eq. (1)). The normalization was performed to reduce the influence of the different lighting conditions to the colour channels. Afterwards, through a first-stage digital image processing, the images were normalized in its green channel (Eq. (2)). This was done in order to improve green colour detection based on the elimination of light and shadow in the images. Then, the ExG (Excess Green) vegetation index, proposed in Woebbecke et al., (1995), was used to perform the initial vegetation segmentation (Eq. (3)). The ExG index gives the differences between the detected light values of green channel and the red and blue channels.

$$R^* = \frac{R}{255};\ G^* = \frac{G}{255};\ B^* = \frac{B}{255} \tag{1}$$

$$r = \frac{R^*}{R^* + G^* + B};\ g = \frac{G^*}{R^* + G^* + B};$$

Fig. 2. Image examples in our dataset: (a) cotton; (b) tomato; (c) velvetleaf; (d) black nightshade.

[1] https://github.com/AUAgroup/early-crop-weed

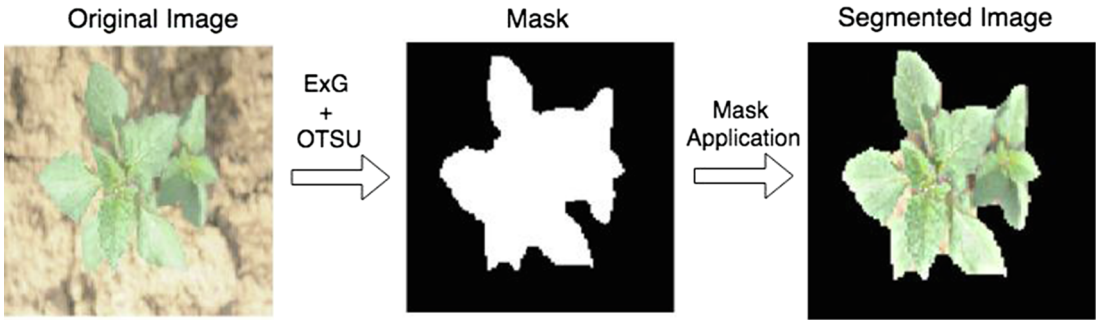

Fig. 3. Example (RGB) image from the dataset, normalized (RGB) image, Vegetation Mask.

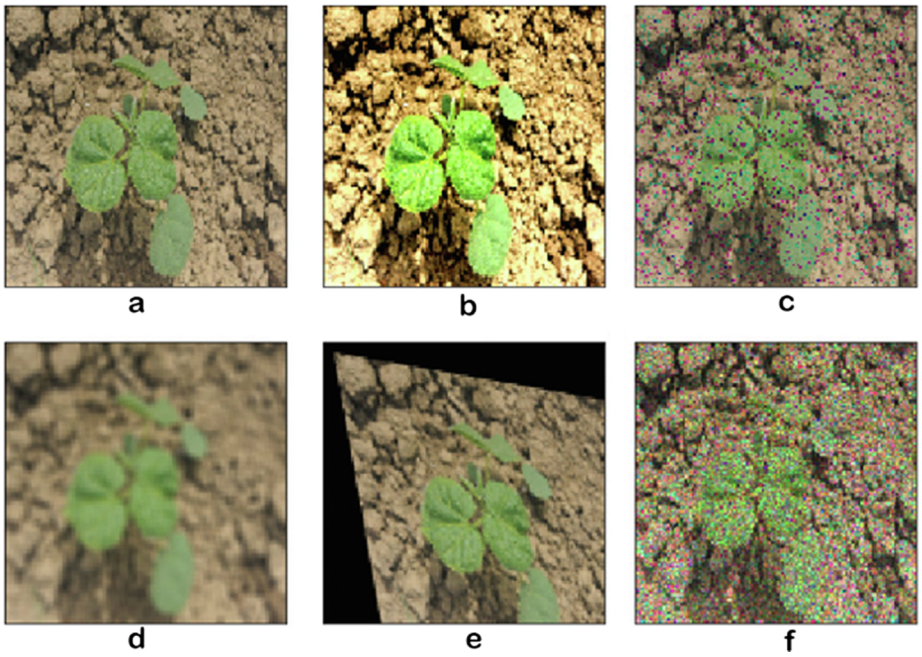

Fig. 4. Cotton image with data augmentation techniques applied: (a) Original; (b) Contrast-shifting; c) Salt-Pepper noise; (d) Blurring; (e) Rotation; (f) Gaussian noise.

$$b = \frac{B^*}{R^* + G^* + B}; \tag{2}$$

$$ExG = 2g - r - b \tag{3}$$

Following, an OTSU thresholding (Otsu, 1979) (this threshold is determined by minimizing intra-class intensity variance) was applied to the grayscale images in order to obtain a binary mask. Finally, the original images were segmented/masked in order to differentiate plants from the soil and other non-plant pixels. This means that the backgrounds were the same for all images and, thus, they could not participate in the determination of plant species.

2.3. Transfer Learning: Fine-tuning models

After the dataset was preprocessed through a plant segmentation method, it was necessary to extract relevant features and classify the resulted images based on them. CNNs can be used as a highly-accurate method for automatic feature extraction and image classification. CNNs work by convolving over images, adjusting their weights and creating a representation of hierarchical self-learned features that is fed into a classic fully-connected neural network. CNNs weights can either be (i) randomly initialised, or (ii) transferred from a pre-trained model in a process called transfer learning. Pre-trained weights usually have useful information and, therefore, a pre-trained model is a much better starting point; especially for extracting features on small image datasets (as is our case, see Table 2). If the original dataset is large enough and general enough, reusing these weights make sense, because the first layers contain the more abstract features and they act as a generic visual model detecting edges and blobs among other simple elements. By this way, we are trying to leverage the knowledge learned by the network from the previous dataset (in this case the ImageNet).

There are two ways to apply transfer learning with pre-trained deep networks. One possibility is to utilize the pre-trained network with the learned weights to obtain features that would be subsequently used in the new problem. Here, the outputs of the network, prior to the last fully-connected layer, constitute features of interest. Another option is to fine-tune the network weights by training the network with the new dataset and works like this: train the weights that are closer to the output and freeze the other layers. Later layers are more likely to change during fine-tuning as they are closer to the output layers which receive larger back-propagated error differential. There is also the possibility of fine-tuning the entire network when the new dataset is large enough. We evaluated both approaches. Fine-tuned learning experiments require a bit of learning, but they are still much faster than learning from scratch (Mohanty et al., 2016). In some cases, they are also more accurate compared to models trained from scratch.

There exist several successful popular pre-trained networks which researchers can use to start building their models. Even though these architectures are based on LeNet-5 (Lecun et al., 1998) (one or more

convolutional layers with fully connected layers on top), each of them has different advantages and scenarios where it is used more appropriately (Canziani et al., 2017). In this work, we evaluated the following pre-trained CNN architectures: VGG16 and VGG19 (Simonyan and Zisserman, 2014) (VGG stands for Visual Geometry Group); Inception (Szegedy et al., 2015); Inception-ResNet (Inc-Res) (Szegedy et al., 2016); Densenet (Huang et al., 2017); Xception (Chollet, 2017) and Mobilenet (Howard et al., 2017).

Although these networks are translation invariant, they are not invariant to another image variations such as rotation. For that reason, to improve generalisation properties and reduce overfitting, in this work we implement data augmentation through random rotation and zooming; Gaussian, Poisson and salt-pepper noise addition, and finally, contrast and brightness shifts. Some examples are shown in Fig. 4. It is important to remark that these transformations were applied only to training data. Validation and test sets were not modified. In this way, we expected that the tricky training had positive consequences with the validation and tests datasets. Resampling techniques were not used, since in crop-based applications the plant distribution is more balanced in contrast to other related research.

Once these architectures extracted the features, it was necessary to train a classifier that discerns crops from weeds. This was achieved by a classical fully-connected network. In a fully connected network, each neuron in a layer is connected to every neuron in the previous layer, and each connection has its own weight. This is a totally general-purpose connection pattern and makes no assumptions about the features in the data. In order to avoid overfitting (i.e., the algorithm learns useless features), we also added dropout layers. Dropout is only applied during training (backward pass), not on making predictions (forward pass). Owing to the correct configuration of these layers is still more of an art than a science, we experimented with three layers under different configurations that will be explained in the Experiments section.

2.4. Classifier

The next step in our proposed method (see Fig. 1) was to replace the fully-connected network by another machine learning model. The reasoning here is that the fully-connected layers were suitable for feature extraction, but better patterns could be found by different “traditional” classifiers. A common practice is to use the output of the very last layer before the flatten operation, the so-called “bottleneck layer”. The bottleneck features retain general knowledge which can be converted in feature vectors. These vectors are used to train some machine learning models such as Gradient Boosting, SVM, Logistic Regression, or Random Forests on top of these features to obtain a classifier that can recognise new classes of images. Our hypothesis was that this classifier replacement allows our weed detector to improve their results in respect to the fine tuning and fully-connected network approach. In this work, we evaluated Logistic Regression, Support Vector machines and Gradient Boosting.

Logistic Regression is a good starting point for many classification problems (as VGGNets are for feature extraction) (Kleinbaum and Klein, 1994; Friedman et al., 2008). SVM is used because, until the rising of deep learning, this algorithm provided state-of-the-art classification models (including the agricultural field (Zhou et al., 2014)) due to their robustness to high dimensionality problems (Cortes et al., 1995). Finally, gradient Boosting has shown very promising results (Dimitrakopoulos et al., 2018; Ustuner et al., 2019). As a non-linear classifier, its computational complexity is one of the main issues while training, but once the model is trained, it can classify in real-time conditions.

2.5. Evaluation

Despite the fact that there are lots of heuristics and best practices, currently, the majority of deep learning design decisions need an empirical approach based on trial-and-error rather than a mathematical/formal justification. Therefore, it is necessary to provide a wide evaluation that allows observing an extensive catalogue of configurations.

The performance was measured with the F_1 score (Eq. (4)). This metric is used as the evaluation metric for classification, where precision is the ratio of correct labels in the classifier output and recall is the ratio of the correct categories respect to the original dataset. Since we had four different classes, it was necessary to do some aggregation for comparison purposes. In our experiments, we used micro-averages. In a multi-class classification setup, micro-average is preferable if there might be class imbalance (i.e. there are many more examples of one class than of other classes. See first row of Table 2). To reduce the volatility of the system, we conducted each experiment 10 times under different random seeds and report the mean for each neural architecture. Furthermore, in order to be considered as a suitable approach, the system must obtain a F_1 score higher than 75% in all the experiments. In other words, we consider useless that in the majority of experiments the classifier obtains a F_1 of 95% and in one of them it only obtains a F_1 of 35%.

$$\mathrm{F1} = 2 * \frac{precision * recall}{precision + recall} \tag{4}$$

Another concern of this work, was the robustness of the classifier and its ability to generalise. For this reason, we measured the difference between the performance in train and test data (train/test diff.) in order to verify that the proposed methodology tends to avoid overfitting (Wong et al., 2016; Cardellino et al., 2016). On the other hand, overfitting is a far too complex topic, and a single metric cannot fully comprehend the various factors that can have an effect; but deeper exploration of this topic has not been taken into consideration in this work.

As application-oriented research, we only use one dataset as a benchmark, avoiding average performance over distinct types of datasets. Once the best crop/weed classifier has been found, we also present a comparison of the performance for each of the main components of the system (i.e., plant segmentation strategy, pre-trained architecture selection and fine-tuning methodology). This part of the evaluation can clarify which are the more promising future research directions.

3. Experimental results

This section shows the results of the experiments performed with the different neural architectures in order to find the most suitable approach for identifying weeds. Moreover, our experiments were designed to show the capabilities of our method and to support our key claims, which are: (i) We can design an end-to-end approach for identifying crop/weed with high performance and low train/test difference; (ii) the use of a traditional shallow classifier on the top of deep features can improve the results of the fully-connected approach. In addition, we analysed the results not only for our specific problem, but to illuminate the nature of learning mechanisms in deep neural architectures and the reasons for their success or failure. This part was in line with the larger concern about how well our experimental results will generalise to other yet unseen related problems. The experiment results were obtained averaging 10 different trials. In each case, a stratified split is performed with 65% of the samples used for training, 15% for validation and 20% for testing.

3.1. Experimental setup

3.1.1. Software and hardware setup

For the implementation of the neural networks, we used Keras 2.1.6[2] (Chollet et al., 2015) with Tensorflow 1.13.1[3] in the back end. The state-of-the-art pre-trained networks included in the Keras core library represent some of the highest performing CNNs on the ImageNet challenge over the past few years. When using Keras as the deep learning framework, each network requires all training images to have the same size. In the weeds identification dataset, we have images of different sizes; therefore, after the segmentation step, we resized the images into a square 128x128. Scaling of the images is good for removing the correlation between image size and physical size of the plants, thus giving the ability to mix images from different datasets, despite being acquired with different cameras at different heights. As an identifier, we used Scikit-learn 0.21.2[4] and for Gradient Boosting, we used XGBoost[5], an algorithm that has recently been dominating applied machine learning for structured or tabular data. All the experiments were run on Ubuntu 18.04 as the OS, on a desktop computer with a 16-core CPU (Ryzen Threadripper 1950x), 64 GB of DDR4 ram, and a GeForce RTX 2080Ti GPU with 11 GB of integrated DDR5 memory. Deep learning relies heavily on GPU, so we also enabled the available CUDA cores of the GPU in favour of processing speed.

3.1.2. Hyperparameter tuning

Optimizing machine and deep learning models is not an exact science. The best architecture, optimization algorithm and hyperparameter settings depend on the dataset. Thus, being able to quickly test several model hyperparameters is imperative in maximizing performance. For that reason, to start the training process, we have done several experiments by grid search to find the best hyperparameters for our proposal. We have experimented with removing or adding layers in the fully-connected network, changing the activation functions, freezing more or fewer layers in the fine-tuning process, decreasing learning rate, etc. Since all these ones interplay tightly with the final performance, the "best" hyperparameters were set as an experimental constant to enable a fair comparison between the architectures and standardize the hyperparameters across all the experiments. Table 3 summarizes the primary hyperparameters that governed neural networks during our experiments. Boldface indicates the best performer in case that various configurations were evaluated.

Table 3
Hyperparameters used in experiments.

Parameter	Value
Fully-Connected Configuration	{2048, 1024, 512, 4}, **{4096, 2048, 512, 4}**
Activation Functions	Relu
Weights Initialization	Glorot Uniform
Base Learning Rate	{0.001, **0.0001**}
Learning Rate Policy	Decreases by a factor of 10 every 30 epochs
Dropout Rate	**0.25**, 0.5
Optimizer	**Adam**, SGD
Batch Size	**16**, 32
Max. Epochs	40, **80**

3.2. Experimental results

3.2.1. Best weed classifier

Table 4 shows the best identification systems according to our performance criteria. Additionally, we have used the notation of "Segmentation:Arch:Transfer:Pooling + Classifier" to refer to particular system configurations. For instance, to refer to the experiment using a combination of Logistic Regression with the VGG19 architecture, which was trained using pooling, plant segmentation and fine-tuning, we use the notation "PS:VGG19:FT:AMP + Log".

The results showed that some architectures such as Densenet, VGG16 and VGG19 had a good performance in this problem. Moreover, these architectures also showed a low train/test diff. On the other hand, other architectures such as Inception or Xception have shown consistently inferior results. There is a factor that appears relevant to obtain a good classifier; Plant Segmentation. While Densenet obtained a good performance without plant segmentation, both VGG19 and VGG16 had good performance when they were combined with this preprocessing step. Finally, it is important to note that Densenet required the complement of the fine-tuning (FT) technique in achieving the best performance.

The "Improv" table column reflects the improvement (positive value) from the fully-connected approach in comparison to the other classifiers (Log, SVM and XGB). All the configurations shown in this table improve the fully-connected approach. However, there are some examples shown in Table 5 where the classifier replacement deteriorates the results. Additionally, this table shows the bad performance achieved by some pre-trained networks that are evaluated in the same conditions as the others. From this contrast, we can conclude that the selection of the incorrect configuration can drift in poor performances.

In Table 6, the best combination between each of the deep architectures and the corresponding classifier is presented. Both fine-tuning and feature-extraction approaches are visualised. It can be observed that in the case of the fine-tuning approach, MobileNet obtained its best performance with SVM, and the same happened with IncRes, Xception, VGG16 and VGG19. On the other hand, taking into account the feature-extraction approach, Log and SVM were usually the best replacements to the fully-connected. XGB only obtained the best performance with Inception (Fine-tuning) and IncRes (Feature-Extraction).

Finally, according to the evaluation criteria, the best configuration is "NPS:Densenet:FT:AMP + SVM". Thus, the complete methodology for weeds/crop identification is formed by the next steps:

1. An RGB image is received.
2. Image is resized to 128X128 pixels.
3. Plant Segmentation is not applied.
4. Feature extraction is made by a fine-tuned Densenet.
5. Plant identification with SVM.

Once, the best systems according to our criteria have been found, in the next sections, we study deeply different components to find out their implication in the final performance and to provide some insights about the design of similar systems.

3.2.2. Best feature Extraction: Fine-Tuning or not?

Fig. 5 shows different points of view when analysing the impact of fine-tuning. Fig. 5a shows the difference between the performance in train and test sets, which showcases the ability of the system to fit the problem and avoid overfitting. Observing the medians, it can be argued that fine-tuning strategy had more low results than the CNN-based feature extraction approach. On the other hand, with fine-tuning there was more sparsity, and high overfitting can also be observed. What it can be concluded is that with both approaches low train/test difference could be achieved. Fig. 5b and c show the F_1 score with the fully-connected approach and with the classifier replacement. In this case, it is clearer that fine-tuning allows obtaining higher performance with a fully-connected classifier. Since most of the features learned by the

[2] https://keras.io/
[3] https://www.tensorflow.org/
[4] https://scikit-learn.org/stable/
[5] https://xgboost.readthedocs.io/en/latest/

Table 4
Best Weed-Crop Classifier Systems. Notation: (N)PS: (No) Plant Segmentation; (N)FT: (No) Fine-Tuning; NP/AMP: No Pooling/Average-Maximum Pooling.

Classifier Configuration	μ-F_1%	Train/test diff.	Improv.
NPS:DenseNet:FT:AMP + SVM	99.29 ± 0.70	0.50 ± 0.40	4.29 ± 1.67
NPS:DenseNet:FT:AMP + Log	99.14 ± 0.64	0.64 ± 0.63	4.14 ± 1.81
NPS:DenseNet:FT:NP + SVM	99.00 ± 0.76	0.79 ± 0.75	5.43 ± 3.77
NPS:DenseNet:FT:NP + Log	99.00 ± 0.75	0.79 ± 0.77	5.43 ± 3.70
PS:VGG19:FT:NP + SVM	98.83 ± 0.69	0.97 ± 0.68	3.00 ± 2.24
PS:VGG19:NFT:NP + SVM	98.50 ± 0.50	1.30 ± 0.47	3.17 ± 2.67
PS:VGG19:FT:NP + Log	98.50 ± 1.26	0.85 ± 0.23	3.17 ± 3.34
PS:VGG16:NFT:NP + Log	98.38 ± 0.86	1.40 ± 0.98	18.18 ± 4.86
PS:VGG16:FT:NP + Log	98.33 ± 1.25	1.47 ± 1.45	1.33 ± 1.05
PS:VGG16:FT:NP + SVM	98.33 ± 0.94	1.42 ± 0.94	2.67 ± 1.11

Table 5
Configurations which did not improve the classifier replacement.

Fine-tuning	μ-F_1%	Improv
NPS:Xception:NFT:AMP + Log	54.89 ± 3.35	−14.34 ± 2.56
NPS:Xception:NFT:NP + Log	57.06 ± 1.07	−12.50 ± 3.20
PS:Inception:FT:NP + Log	37.34 ± 1.30	−9.40 ± 3.28
PS:Inception:FT:NP + XGB	42.32 ± 2.39	−5.60 ± 5.42
PS:Xception:FT:AMP + Log	36.50 ± 2.12	−4.47 ± 9.71

Table 6
Comparison of the best combination between deep architectures (with a fine-tuning and feature extraction approach) and classifiers.

Fine-tuning	μ-F_1%	Feature-Extraction	μ-F_1%
Mobilenet + SVM	98.50 ± 1.75	Mobilenet + Log	98.00 ± 1.23
IncRes + SVM	91.67 ± 9.41	IncRes + XGB	79.00 ± 0.79
Xception + SVM	88.20 ± 11.06	Xception + SVM	96.25 ± 2.11
Inception + XGB	88.00 ± 5.23	Inception + SVM	94.50 ± 1.73
VGG16 + Log	98.33 ± 1.25	VGG16 + Log	98.38 ± 0.86
VGG19 + SVM	98.83 ± 0.69	VGG19 + Log	98.50 ± 1.26
Densenet + SVM	99.29 ± 0.56	Densenet + SVM	98.00 ± 0.68

frozen networks were not useful for distinguishing plants, this result has complete sense. In fact, features learned from pre-trained deep neural models on a large image dataset without fine-tuning may not generalize well in agricultural images. However, it is also important to observe that due to the sparsity of the distributions, it exists the possibility of obtaining low performances. Additionally, when classifier replacement is used, the superiority is not so clear (more sparsity but also the highest maximum). Therefore, it can be argued that fine-tuning was important but not definitive.

3.2.3. *Plant segmentation: Remove background or not?*

Fig. 6 provides some light to the utility of using plant segmentation or not. As it was explained previously, it is supposed that plant segmentation or background removal should be avoided when using deep learning, however, some related works show that this is not always the case and segmenting the plant before training has consequences in the system performance. Fig. 6a shows that removing the background did not have a better fit to the identification problem. In fact, not removing the background shows a lower median and a less sparse distribution. Fig. 6b shows that plant segmentation did not improve results, but it is not so clear when classifier replacement is used (Fig. 6c)). Therefore we can argue that taking into account the experimental constraints, plant

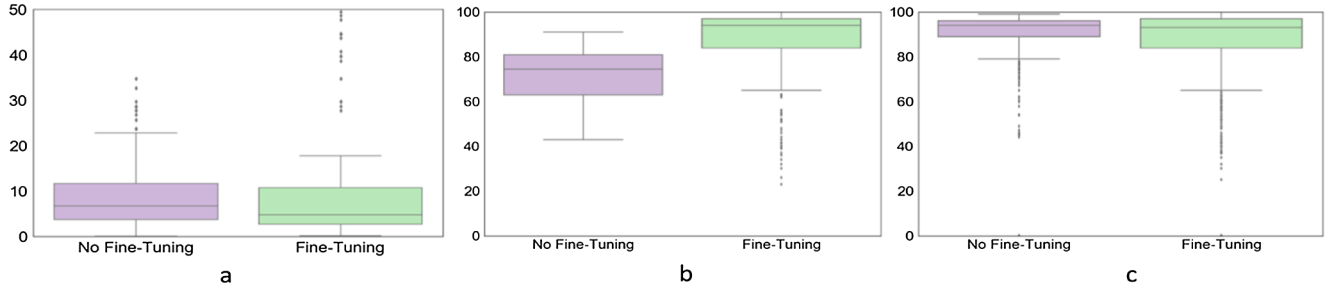

Fig. 5. Transfer-Learning analysis: (a) train/test diff. (b) fully-connected F_1 score; (c) classifier replacement F_1 score.

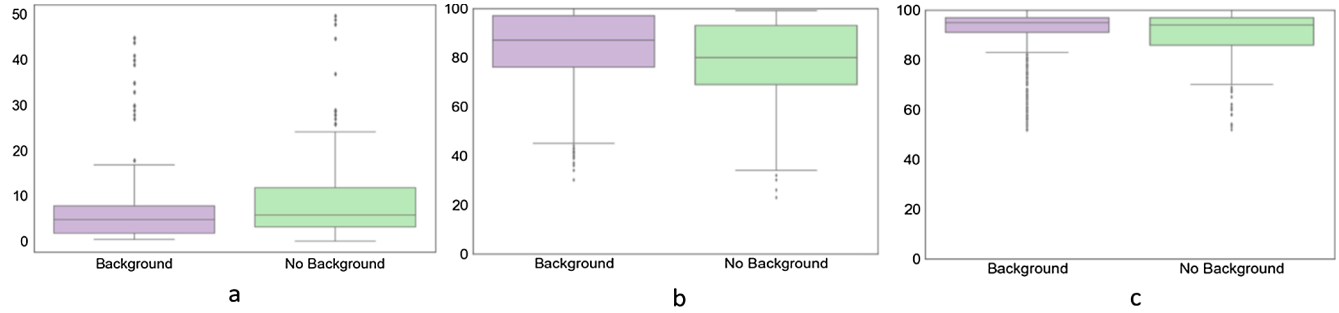

Fig. 6. Plant-Segmentation analysis: (a)train/test diff. (b) fully-connected F_1 score; (c) classifier replacement F_1 score.

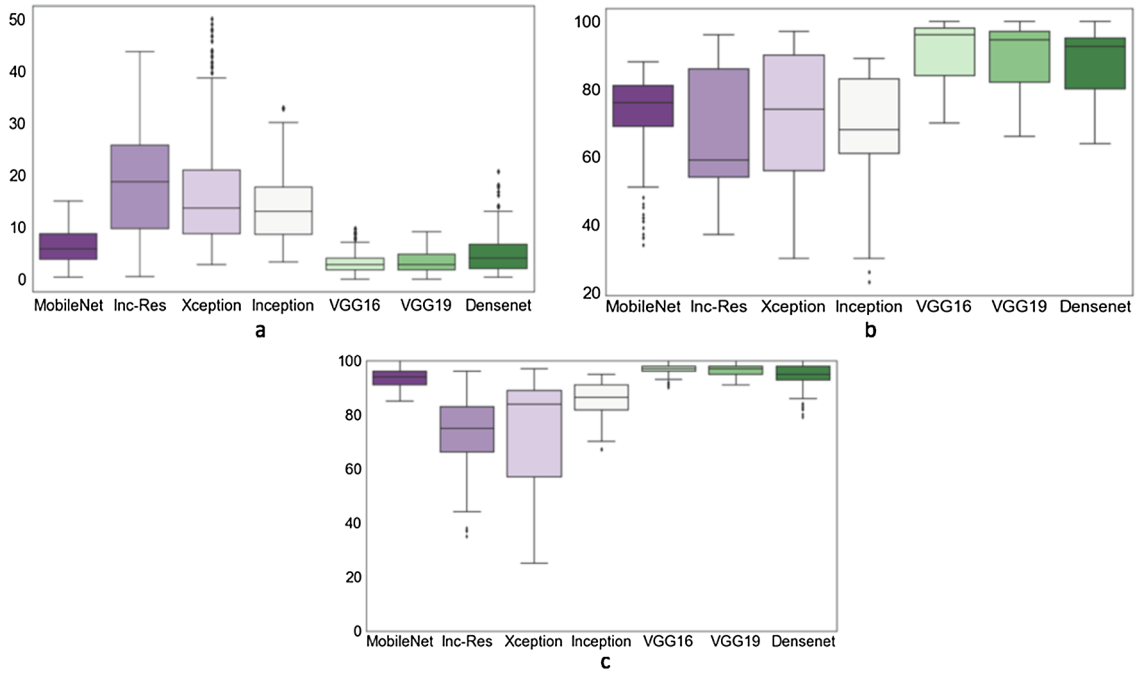

Fig. 7. Pre-trained networks analysis: (a) train/test diff. (b) fully-connected F_1 score; (c) classifier replacement F_1 score.

segmentation is an important step when the classifier replacement technique is not applied.

3.3. Pre-trained networks analysis

In the current research, we have studied several pre-trained architectures under different configuration in order to find the best system to implement the crop/weed classifier. In Fig. 7, the different architectures are opposed to verify the superiority or not among them. It can be observed that VGG16, VGG19 and Densenet are clearly the best choice for our proposal. They were the best architectures for reducing train/test diff. (Fig. 7a) and they also obtained the best performances: both with a fully connected approach and replacing the classifier by another one. This result was expected due to the results presented in Table 4. Mobilenet also showed potential, but clearly lesser to the VGGNets and Densenet architectures. Finally, more complex networks such as Inception, Xception and Inception-Resnet obtained very poor results.

4. Discussion

In this work, a crop/weed classifier has been developed, which is an important step to support crop protection and prevent weed dispersal, due to the more effective weed management, when expert knowledge is unavailable. The methodology was composed of several steps which can adopt different configurations. The main objective was to find the configuration that offered the best performance according to our evaluation criteria. In this case, the system "NPS:DenseNet:FT:AMP + SVM" achieved the best performance with a F_1 score of 99.29% on average, besides a low difference between train/test performance, which could be considered as a quite close end-to-end system. This performance was in accordance with state-of-the-art research in weed detection (Bakhshipour and Jafari, 2018; Milioto et al., 2018; Kounalakis et al., 2019). This system was formed by some of the configurations that have shown interesting properties for this research. NPS (No Plant segmentation) was important to avoid overfitting; Densenet (alongside VGG19 and VGG16) was the pre-trained architecture with the best performance; and FT (Fine-Tuning) has obtained consistently good results. This correlates with the results obtained by Suh et al., (2018), where VGG19 also obtained an outstanding performance of 98.7%.

If a fully-connected approach is used (Fig. 7b and 7c), we suggest fine-tuned CNN models. However, if a different classifier is used on top of the deep network, no-fine-tuning can also obtain good results. Regarding the "traditional" classifiers, SVM and Logistic regression obtained the best performances, which correlates with the work by Kounalakis et al., (2019), where a combination of CNN-based feature extraction with the L2regLogReg classifier yielded the best and most balanced detection results. Additionally, replacing the fully-connected network and training a different classifier with the deep extracted features could sometimes (67.06% of the experiments) improve the performance (See Figs. 5, 6 and 7).

On the other hand, it is important to remark that due to the "No free lunch theorem" (Wolpert and Macready, 1995), as new photos are added to the current dataset, new approaches should be evaluated because the statistical nature of the data could be changed and a new configuration could obtain the best performance. The performance deterioration due to changes in the dataset has been previously reported by different authors (e.g., Mohanty et al. 2016; Barbedo, 2018; Ferentinos, 2018). Since some positive thoughts can be extracted from the train/test difference in performance (see Table 4), we could argue that some architectures are quite robust to the change in the input data. But, current experiments were performed with the same database and the performance deterioration could increase exponentially with a different dataset.

It is of high interest that that some state-of-the-art architectures showed quite low performance (see Fig. 7). This may be explained by the initial resizing stage of the photo, which could decrease the network performance. Actually, each of the pre-trained models is optimized with specific spatial tensor sizes of either 299 × 299 (Inception, Xception, and InceptionResNet) or 224x224 (VGGNets, MobileNet and DenseNet); and we finally chose a size of 128x128 due to a "ResourceExhaustedError" when trying larger image sizes. It can be observed that VGGNets, MobileNet and DenseNet are the networks which are optimized for smaller pictures (224x224), and, indeed, they were the ones with the best performance in the experiments (See Fig. 7c)). Another constraint of the proposed methodology was that only the deep features extracted by one network were used; an ensemble of several

deep features feeding the traditional classification model could overcome the current performance.

Regarding to the system deployment, there are mainly two ways: (i) the whole system embedded in the smartphone; (ii) a distributed approach, where the neural network is behind a web service. Both of them have advantages and disadvantages. The standalone approach does not need an internet connection and, thus, it seems to be the right configuration for a safe real-time classification; whereas the distributed one can be updated easily with new algorithm developments that have obtained better results when new data is acquired. With the standalone deployment, there would be highly valued that the neural network is "small" enough. It should be noted that although fine-tuned VGG19 achieved better performance, it is memory-consuming and therefore may not be suitable for embedded devices. For embedded devices, some more memory-saving models such as Mobilenet will be a better choice. However, in the presented experiments Mobilenet has shown an overall irregular behaviour and a poor performance in many situations; therefore, more experiments with different configurations on it should be done.

Another objective of the research was to answer some general questions about the abilities of the different system configurations to favour better or worse results in order to provide some intuitive hints for related research works. Empirical results showed that plant segmentation tends to reduce overfitting, whereas a fine-tuning strategy had more chance to obtain a better performance. However, any of these configuration setups could not obtain a good performance by themselves. The correct combination of each stage is what obtains the necessary performance to develop a suitable weed identifier.

5. Conclusions

In this study, a novel vision-based classification system for identifying weeds in crop fields, by exploiting pre-trained deep neural networks, was examined. The study focused on volunteer tomato and volunteer cotton, common weeds for tomato plants and cotton plants, respectively. We have performed an extensive evaluation of the algorithms using real data. The best crop/weed identifier has obtained a promising performance of 99.29% F_1 score besides a low tendency to overfitting. Moreover, result analysis has shown the strengths and weaknesses of using fine-tuning, plant segmentation and classifier replacement.

Another contribution is the dataset, which we have also used for evaluation purposes. The dataset contains 504 RGB images taken from approximate height of 1 m. This dataset is a good starting point for initial development of machine vision algorithms for weed detection from RGB images and is one of the rarely available public datasets for this purpose.

As future work, we plan to validate our approach with additional datasets such as PlantVillage[6] or Plant Seedlings Dataset[7] in order to verify the generality of our approach and our findings. We will also study the reasons why some pre-trained architectures that obtained good performance in related work, do not behave as expected with our dataset; and whether by using their default input size we can overcome this issue. Additionally, more strategies to prevent overfitting such as finer regularization or dropout layers will be deeply studied. Furthermore, since more pictures will be added to the provided dataset, more experiments will be performed in order to always deploy the best weed identification system.

Finally, we conclude that this image-based technology can play an important role in detecting and recognising weeds. The use of these applications could help in weed identification at early growth stages at an unprecedented scale. As result, it could increase herbicide efficacy due to their application at more susceptible stage and contribute to food security and environmental protection by reducing the amount of herbicides used.

[6] https://github.com/spMohanty/PlantVillage-Dataset/tree/master/raw/

[7] https://vision.eng.au.dk/plant-seedlings-dataset/

CRediT authorship contribution statement

Borja Espejo-Garcia: Investigation, Software, Writing - original draft. **Nikos Mylonas:** Data curation, Resources, Writing - original draft. **Loukas Athanasakos:** Data curation, Resources, Writing - original draft. **Spyros Fountas:** Supervision, Writing - review & editing. **Ioannis Vasilakoglou:** Supervision, Writing - review & editing.

Acknowledgement

The work of Borja Espejo-Garcia has been partially supported by the Government of Aragon and the European Social Fund through the grant number [C38/2015]. This work has also been supported by the Spanish Government (project TIN2017-88002R); and Aragon Government and EU FEDER program project T59_17R. Field surveys and agronomic support has been kindly sponsored by Corteva Agriscience™.

References

Bakhshipour, A., Jafari, A., 2018. Evaluation of support vector machine and artificial neural networks in weed detection using shape features. Comput. Electron. Agric. 145, 153–160.

Barbedo, J.G.A., 2018. Factors influencing the use of deep learning for plant disease recognition. Biosyst. Eng. 172, 84–91.

Bengio, Y., 2012. Deep learning of representations for unsupervised and transfer learning. J. Machine Learning Res. 17–37.

Bock, C.H., Poole, G.H., Parker, P.E., Gottwald, T.R., 2010. Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. Biosyst. Eng. 172, 84–91.

Cardellino, C., Alemany, L.A., Teruel, M., Villata, S., Marro, S., 2016. Convolutional ladder networks for Legal NERC and the impact of unsupervised data in better generalizations. The Thirty-Second International Florida Artificial Intelligence Research Society Conference (FLAIRS-32).

Canziani, A., Paszke, A., Culurciello, E., 2017. An Analysis of Deep Neural Network Models for Practical Applications. arXiv:1605.07678.

Chollet, F., and others, 2015. Keras. https://keras.io.

Chollet, F., 2017. Xception: deep learning with depthwise separable convolutions. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR. 2017;2017-January 1800–1807. arXiv:1610.02357v3.

Cortes, C., Vapnik, V., 1995. Support-Vector Networks. Machine Learning; 20(3):273–297. arXiv:1011.1669v3.

Dimitrakopoulos, G.N., Vrahatis, A.G., Sgarbas, K., Plagianakos, V., 2018. Pathway analysis using xgboost classification in biomedical data. ACM International Conference Proceeding Series:1–6.

European Crop Protection Agency (ECPA), 2017. European Crop Protection: With or without pesticides? URL: https://www.ecpa.eu/with-or-without.

Ferentinos, Konstantinos P., 2018. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. 145, 311–318. https://doi.org/10.1016/j.compag.2018.01.009.

Food and Agriculture Organization, 2013. The state of food and agriculture: food systems for better nutrition. volume 2, 2009. arXiv:978-92-5-105980-7.

Friedman, J., Hastie, T., Tibshirani, R., 2008. The Elements of Statistical Learning.

Gebbers, R., Adamchuk, V.I., 2010. Precision agriculture and food security. Science 327 (5967), 828–831.

Haug, S., Michaels, A., Biber, P., 2014. Plant classification system for crop/weed discrimination without segmentation. In: IEEE Winter Conference on Applications of Computer Vision, pp. 1142–1149.

Haug, S., Ostermann, J., 2018. A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. In: In: European Conference on Computer Vision, pp. 1–12.

Howard, A.G., Wang, W., Zhu, M., Chen, B., Kalenichenko, D., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.arXiv:1704.04861v1.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2017, pp. 2261–2269 arXiv:1608.06993v5.

Kitzes, J., Wackernagel, M., Loh, J., Peller, A., Goldfinger, S., Cheng, D., Tea, K., 2008. Shrink and share: humanity's present and future ecological footprint. Philos. Trans. Royal Soc. B: Biolog. Sci. 363 (1491), 467–475.

Kleinbaum, D.G., Klein, M., 1994. Logistic Regression: A self-learning text. Springer-Verlag, N.Y.

Kounalakis, T., Triantafyllidis, G.A., Nalpantidis, L., 2016. Weed recognition framework for robotic precision. farming. 2016 IEEE International Conference on Imaging Systems and.

Kounalakis, T., Triantafyllidis, G.A., Nalpantidis, L., 2017. Image-based recognition

framework for robotic weed control systems. Multimedia Tools Appl 77 (8), 9567–9594.
Kounalakis, Tsampikos, Triantafyllidis, Georgios A., Nalpantidis, Lazaros, 2019. Deep learning-based visual recognition of rumex for robotic precision farming. Comput. Electron. Agric. 165, 104973. https://doi.org/10.1016/j.compag.2019.104973.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, pp. 1097–1105 arXiv:1102.0183.
Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.
Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. IEEE.
Lottes, P., Hörferlin, M., Sander, S., Stachniss, C., 2016. Effective vision-based classification for separating sugar beets and weeds for precision farming. J. Field Rob. 34 (6), 1160–1178.
Lottes, P., Khanna, R., Pfeifer, J., Siegwart, R., Stachniss, C., 2017. UAV-Based crop and weed classification for smart farming. Automation (ICRA).
McCool, C., Perez, T., Upcroft, B., 2017. Mixtures of lightweight deep convolutional neural networks: applied to agricultural robotics. IEEE Rob. Autom. Lett. 2 (3), 1344–1351.
Milioto, A., Lottes, P., Stachniss, C., 2018. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. Proc. IEEE Int. Conf. Robot. Autom arXiv:arXiv:1709.06764v2.
Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. frontiers. Plant Sci. 7 September 1–10. arXiv:1604.03169.
Oerke, E., 2006. Crop losses to pests. J. Agri. Sci. 31–43.
Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybernet. 9 (1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076.
Potena, C., Nardi, D., Pretto, A., 2016. Fast and accurate crop and weed identification with summarized train sets for precision agriculture. Adv. Intelligent Syst. Comput 531, 105–121.
Rizzardi, M., Fleck, N., 2004. Methods of quantification of weeds and soybean leaf covers. Ciencia Rural 34 (1), 13–18.
Simonyan, K., Zisserman, A., 2014. very deep convolutional networks for large-scale image recognition. ICLR 2015, 1–14 arXiv:1409.1556.
Suh, H.K., Ijsselmuiden, J., Hofstee, J.W., van Henten, E.J., 2018. Transfer learning for the classification of sugar beet and volunteer potato under field conditions. Biosyst. Eng. 174, 50–65.
Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2016. Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence, pp. 4278–4284.
Szegedy, C., Vanhoucke, V., Loffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2015-Decem:2818–2826:arXiv:1512.00567v3.
United Nations (UN), Department of Economic and Social Affairs, Population Division, 2019. World Population Prospects 2019, Online Edition.
Ustuner, M., Sanli, F.B., Abdikan, S., Bilgin, G., Goksel, C., 2019. A booster analysis of extreme gradient boosting for cropclassification using PolSAR imagery. 8th International Conference on Agro-Geoinformatics.
Wang, G., Sun, Y., Wang, J., 2017. Automatic image-based plant disease severity estimation using deep learning. Comput. Intelligence Neurosci. 2017, 8.
Woebbecke, D., Meyer, G., Von Bargen, K., Mortensen, D.A., 1995. Color indices for weed identification under various soil, residue, and lighting conditions. In: Transactions of the ASAE. American Society of Agricultural Engineers, pp. 259–269.
Wolpert, D.H., Macready, W.G., 1995. No Free Lunch Theorems for Optimization. Technical Report.
Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D., 2016. Understanding data augmentation for classification: when to warp? International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016.
Zhou, R., Kaneko, S., Tanaka, F., Kayamori, M., Shimizu, M., 2014. Image-based field monitoring of *Cercospora* leaf spot in sugar beet by robust template matching and pattern recognition. Comput. Electron. Agric. 108, 58–70.
Zimdahl, R.L., 2018. Fundamentals of weed science, fifth ed. Academic Press Elsevier Inc.