Fast and Accurate Crop and Weed Identification with Summarized Train Sets for Precision Agriculture

Ciro Potena, Daniele Nardi, and Alberto Pretto

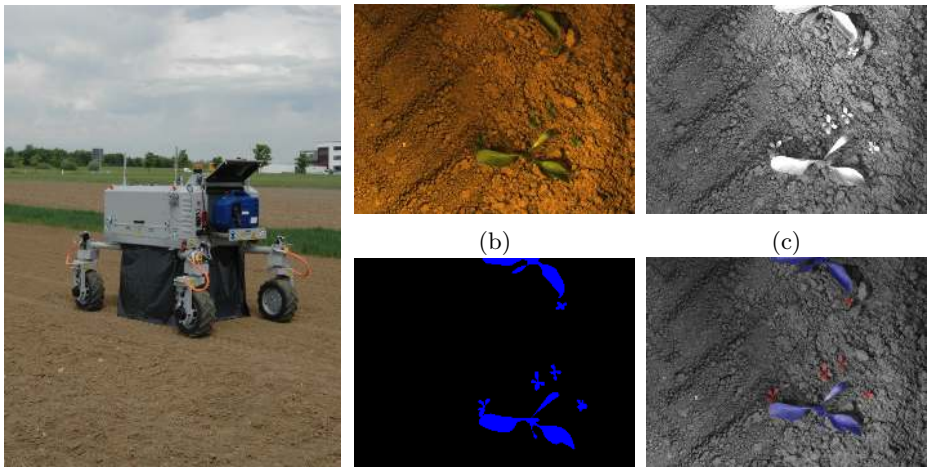
Department of Computer, Control and Management Engineering
Sapienza University of Rome
Via Ariosto 25, 00185, Rome, Italy
{potena, nardi, pretto}@dis.uniroma1.it

Abstract. In this paper we present a perception system for agriculture robotics that enables an unmanned ground vehicle (UGV) equipped with a multi spectral camera to automatically perform the crop/weed detection and classification tasks in real-time.
Our approach exploits a pipeline that includes two different convolutional neural networks (CNNs) applied to the input RGB+near infra-red (NIR) images. A lightweight CNN is used to perform a fast and robust, pixel-wise, binary image segmentation, in order to extract the pixels that represent projections of 3D points that belong to green vegetation. A deeper CNN is then used to classify the extracted pixels between the crop and weed classes.
A further important contribution of this work is a novel unsupervised dataset summarization algorithm that automatically selects from a large dataset the most informative subsets that better describe the original one. This enables to streamline and speed-up the manual dataset labeling process, otherwise extremely time consuming, while preserving good classification performances.
Experiments performed on different datasets taken from a real farm robot confirm the effectiveness of our approach.

Keywords: agriculture robotics, classification, segmentation, convolutional neural networks

1 Introduction

The application of autonomous robotics to precision agriculture is gaining a great attention in the research community, also thanks to the positive impacts that it may have in food security, sustainability and reduction of chemical treatments. In this work, we focus on applications that aim to reduce the amount of herbicides used to control weeds by means of autonomous robots that perform selective spraying or mechanical removal of accurately detected weeds. The robot in this case should autonomously detect and distinguish between crop and weeds inside the field, using only its own perception system.



(a) (b) (c) (d) (e)

Fig. 1: (a) One of the BOSCH Bonirob employed to acquire the datasets used in the experiments; (b),(c) An example of an RGB+NIR images couple provided by the multispectral camera mounted on the robot; (d) The output segmented image obtained using our vegetation detection algorithm: blue pixels represent projections of 3D points that belong to green vegetation; (e) The results of the proposed pixel-wise classification algorithm: pixels that belong to crop are highlighted in violet, pixels that belong to weeds are highlighted in red.

In this work we present a robust and efficient weed identification system that leverages the effectiveness of convolutional neural networks (CNNs) in both the detection and classification steps. Our system takes as input 4-channels RGB+NIR images (e.g., Fig. 1(b),(c)), provided by a multi spectral camera mounted on a farm robot (e.g., Fig. 1(a)) that autonomously monitors the crop and can apply selective weed treatments. The weed identification task includes, before plant classification, a plant detection step. Detection is generally a more challenging and time consuming task compared with classification, since it may require an exhaustive search in the whole image, with variable bounding boxes sizes. In the context of green plants the detection task can be simplified by exploiting the Normalized Difference Vegetation Index (NDVI) [20], extracted from the RGB+NIR images: NDVI enables to obtain a simple, fast and pixel-wise segmentation between green vegetation and soil (e.g., [9, 15]). Unfortunately, being threshold-based, this technique is not robust against illumination changes and different soil conditions: a careful tuning of the threshold and an outlier removal process are necessary to get a good segmentation [15]. To overcome these limitations, in this work we propose to combine the NDVI based segmentation with a trained lightweight CNN (that we call *sNet* in the following) that takes as input small patches of the RGB+NIR images. The idea is to use a very conser-

vative threshold in order to select trough the NDVI most of the true positive pixels (i.e., pixels that represent vegetation). The CNN is then used to validate each selected pixel, pruning most of the false positives (e.g., Fig. 1(d)). We will experimentally show that this hybrid technique outperforms the NDVI based segmentation, while preserving a good computational speed.
Pixels marked as vegetation in the segmentation step are then processed with a deeper 3-classes CNN (that we call *cNet* in the following) in order to recognize the category (crop, weeds or soil). Despite we are processing only pixels classified as vegetation in the previous step, we found that including also the class '*soil*' in the *cNet* CNN helps to prune at no cost the remaining false positives not detected by the *sNet* CNN. In order to meet the real-time constraints required by our system, i.e. we need the classification results within one second from the image acquisition time[1], we also propose to employ a *blob-wise* voting scheme, where blobs are connected regions extracted from the segmentation mask. We will experimentally show that: (a) our classification stage achieves state-of-the-art results; (b) the pipeline composed by the two sequential CNNs (*sNet* + *cNet*) obtains similar results if compared with a single *cNet*, but with a considerable gain in speed.

In the last part of this work we address a relatively new problem that we call *unsupervised dataset summarization*. It is well known that CNNs to be effective require large manual labeled training datasets [21]. Unfortunately, plant identification requires a challenging and extremely time consuming per-pixel labeling process. The proposed idea is to reduce the size of the dataset *before* the manual labeling stage, in order to streamline and speed-up the manual dataset labeling process while preserving good classification performances. We propose an algorithm that automatically selects a subset of K images that contain the most informative features over the N images of the whole dataset, $K \ll N$, in order to summarize in the best possible way the original dataset. The labeling process will then involve only these K images. Our features based subsets selection method is different from other max-relevance and min-redundancy feature selection methods (among others, [18, 25]) since it is *unsupervised*, i.e. it does not require the labels as input. We formulate the unsupervised dataset summarization problem as a combinatorial optimization problem, using as reward a submodular set function inspired by the coverage set functions used in text document summarization problems [14]. We will show that our dataset selection algorithm outperforms in all the experiments both the random dataset selection and the supervised manual selection strategies.

2 Related Work

The problem of plant classification can be considered an instance of the so called *fine-grained visual classification* (FGVC) problem, where the purpose is for in-

[1] In our setup, one second represents a resonable time constraint in order to enable the robot to actively remove the weeds as soon as they are detected.

stance to distinguish between species of animals, models of cars, etc. FGVC problems are intrinsically difficult since the differences between similar categories (in our case, plant species) are often minimal, and only in recent works the researchers obtained noteworthy results (e.g., [17, 27]).

Early works in plant classification faced this problems using features extracted by co-occurrence matrices (CCM) from hue, saturation and intensity color space [22] or morphological and color features as input of a Fuzzy classifier [10]. Burks *et al.* [5] proposed to use CCM texture statistics as input variables for a back-propagation (BP) neural network for weed classification. Borregaard *et al.* [4] used two spectrometers covering both visible and near-infra-red, in order to record reflectance spectra in the wavelength range 6601060 nm. The final crop-weed discrimination accuracy is in this case up to 90%. Feyaerts and van Gool [8] presented a performances of a classifier based on multispectral reflectance in order to distinguish the crop from weeds. The best classifier, based on neural networks, reached a classification rate of 80% for sugar beet plants and 91% for weeds.

More recently, Tellaeche *et al.* [24] proposed an automatic approach for detection and differential spraying of weeds. The captured images are segmented into cells, for each cell two area-based values are computed using crop, weed and soil coverage measurements, a Bayesian decision making framework is finally exploited to decide which cells have to be sprayed. Cells are also used in Aitkenhead *et al.* [2], where for each cell the classification is done by a pre-trained neural network. Hussin *et al.* [7] used shape and color features, the former extracted by the Scale Invariant Feature Transform (SIFT), the latter by the Grid Based Color Moment (GBCM). All the extracted features from the test images are then matched by Euclidean Distance with the ground truth, reaching an accuracy of 87,5%. In Haug *et al.* [9] a Random Forest (RF) classifier was proposed. It uses a large number of simple features extracted from a large overlapping neighborhood around sparse pixel positions. This approach achieves strong classification accuracies, due to its ability of discriminating also crops that are very similar to weeds. This approach has been improved in Lottes *et al.* [15] by extending the features set and including a relative plant arrangement prior that helps to obtain better classification results.

Other approaches rely on leaf classification and/or segmentation in order to detect the plant species. The leaf classification problem in complicated background has been addressed in Wang *et al.* [26], where leaf images are segmented using morphological operators, shape features are extracted and used in a moving center hyper-sphere classifier to infer plant species. Kumar *et al.* [12] presented an automatic plant identification application called Leafsnap: the proposed algorithm starts from segmented images of leaves and it exploits curvature features compared with a given database to extract the best match, while using a binary classifier on global image signatures as a validity test. Deformable leaf models and morphology descriptors have been exploited in [6] to cover the variety of leaf shapes. Very recently Han Lee *et al.* [13] presented a leaf-based plant classification system that uses convolutional neural networks to automatically

learn suitable visual features. Also Reyes *et al.* [19] used CNN for fine-grained plant classification: they used a deep CNN with the architecture proposed by Krishevsky *et al.* [11], first initialized to recognize 1000 categories of generic objects, then *fine-tuned* (i.e., specialized) for the specific task to recognize 1000 possible plant species.

The contribution of this work is a visual detection strategy that allows an UGV to autonomously detect crops and weeds in agricultural field environments. The proposed approach makes use of a multispectral camera as input and two CNNs to obtain accurate classification performances in different growth stages. A further contribution is our novel unsupervised dataset summarization algorithm that allows to reduce a large dataset into a smaller one with similar information properties. Selecting smaller training sets with this approach permits to boost up the labeling phase while preserving a good classification accuracy.

3 Vision-Based Plant Classification

3.1 Vegetation Detection

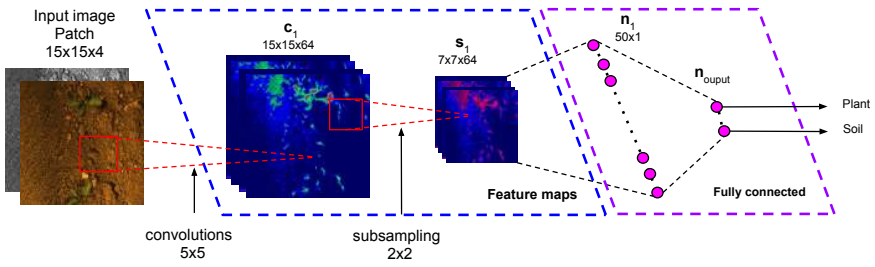


Fig. 2: Architecture of the *sNet* CNN.

The goal of the vegetation detection task is to discriminate in the RGB+NIR images between pixels that represent projections of 3D points that belong to green vegetation and the other pixels. This process enables to simplify and speed up the subsequent plant detection and classification tasks.
Due to the photosynthesis, healthy green plants absorb more solar energy in the visible spectrum, causing a low reflectance level in the RGB channels. Similarly, the reflectance of the near-infra-red spectrum is affected by the same phenomena with opposite results and, as a direct consequence, with a low reflectance level in the NIR channel.

A well known indicator that is used to measure the reflectance properties of the plants is the Normalize Difference Vegetation Index (NDVI) [20], which is calculated as follows for each pixel (u, v):

$$\mathcal{I}_{NDVI}(u,v) = \frac{\mathcal{I}_{NIR}(u,v) - \mathcal{I}_{R}(u,v)}{\mathcal{I}_{NIR}(u,v) + \mathcal{I}_{R}(u,v)} \tag{1}$$

where $\mathcal{I}_R(u,v)$ and $\mathcal{I}_{NIR}(u,v)$ stand for the spectral reflectance measurements taken from the R channel (visible red) and from the near-infrared channel, respectively. The vegetation detection task is typically solved by means of a thresholding operation on the NDVI image: a pixel (u,v) is classified as vegetation if $\mathcal{I}_{NDVI}(u,v) > th_V$, with th_V a fixed threshold. Unfortunately, a single threshold usually is not robust against illumination changes and different soil conditions, even inside a single image. To address this problem, our idea is to combine the NDVI with a lightweight convolutional neural network. We first perform a thresholding operation on the NDVI using a conservative threshold, that allows to preserve most of the pixels that belong to vegetation. For each pixel classified as vegetation, we exploit a trained CNN applied to a 15×15 pixels 4 channels patch around the pixel. This network (*sNet*, Fig. 2) includes a single convolutional layer with rectified linear unit (ReLU) activation function, followed by a max pooling layer and a local response normalization step. We set both strides to 1 in the convolutional layer and both strides to 2 in the pooling layer, where a max pool operator is applied to 2×2 patches. The normalized neurons provided as output from the convolutional and pooling layers are used as inputs for a fully connected layer. The final neurons are then fully connected to the output labels '*plant*' (i.e., vegetation) and '*soil*' (i.e., not vegetation), that are normalized through a softmax layer. The architectural choices made for this CNN represent a good experimental trade-off between the sake of efficiency and the segmentation performances (see Sec. 5.2).

3.2 Pixel-Wise Crop/Weed Classification

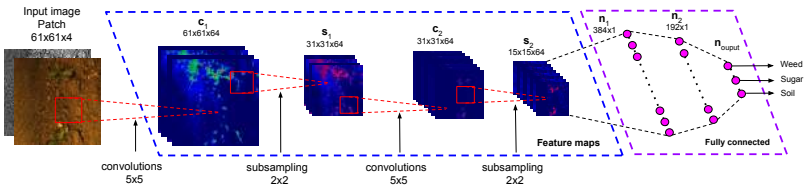


Fig. 3: Architecture of the *cNet* CNN.

The detection system described so far provides an accurate vegetation mask of the input image. Pixels that belong to vegetation need now to be classified between crop and weeds. In this plant classification task there are a lot of possible error sources, among others the similarity between plant species and the partial overlapping between different plants. In order for the network to learn more

Algorithm 1: Blob-Wise Crop/Weed Detection and Classification

Data: The input image $\mathcal{I}$ and a conservative threshold th_{NDVI} for the NDVI.
Result: A set of classified blobs $\mathbf{B}_c$
`/* Compute the vegetation mask` $\mathcal{I}_v$ `*/`
foreach $(u, v) \in \mathcal{I}$ **do**
  $\mathcal{I}_v \leftarrow$ 'soil';
  if $\mathcal{I}_{NDVI}(u, v) \geq th_{NDVI}$ **then**
    if *sNet*$(u, v) =$ *'plant'* **then**
      $\mathcal{I}_v \leftarrow$ '*plant*';
    end if
  end if
end foreach
Extract from $\mathcal{I}_v$ a set of blobs (i.e., connected regions) $\mathbf{B} = \{b_i, \ldots, b_n\}$ of pixel classified as '*plant*';
$\mathbf{B}_c \leftarrow \{\}$;
foreach $b_i \in \mathbf{B}$ **do**
  `/* We randomly sample a number of pixel from the blob, where` s `depends on the blob size` $|b_i|$ `*/`
  Sample s pixel $(u, v)_j$ from the blob b_i, $s < |b_i|$;
  Classify each pixel $(u, v)_j$ using *cNet*;
  if *The majority of the s pixels have been classified as 'sugar'* **then**
    $\mathbf{B}_c \leftarrow \mathbf{B}_c \cup \{b_i, sugar\}$;
  else if *The majority of the s pixels have been classified as 'weed'* **then**
    $\mathbf{B}_c \leftarrow \mathbf{B}_c \cup \{b_i, weed\}$;
  end if
end foreach

specific features that help to disambiguate in these challenging conditions, we move to a slightly deeper network with input patches of 61×61 pixel over the 4 RGB+NIR channels and, accordingly, an higher number of output neurons for every layer. The final network *cNet* (Fig. 3) includes two convolutional layers with ReLU activation function, each followed by a max pooling layer and a local normalization layer. As in the *sNet*, both the max pooling layers of *cNet* operate on 2×2 patches with strides of 2 pixels. The normalized feature maps are then used as inputs for two fully connected layers before passing through a softmax activation function.
Despite the *cNet* processes only pixels classified as vegetation by *sNet* (i.e, pixel classified as "plant"), we still keep the class '*soil*' as a possible output of *cNet*. We experimentally found that this helps pruning at no cost the remaining false positives not detected by the *sNet* CNN.

3.3 Blob-Wise Crop/Weed Classification

The plant detection and classification pipeline presented in the previous section provide state-of-the-art results, but it still suffers from some limitations: (a) A

pure pixel-wise approach can lead to the detection of false positive plants composed by very few mis-classified pixels; (b) Differently from *sNet*, *cNet* does not meet the real-time constraints required by our system.

In order to address these problems, we propose to employ a blob-wise based voting scheme that speeds-up the processes while removing most of the small false positives plants. The pseudo-code of the proposed method is reported in Algorithm 1: we first compute the vegetation mask $\mathcal{I}_v$ as described in Sec. 3.1 (lines 1-8), we extract all the connected regions whose pixels are classified as '*plant*' (line 9) and, finally, we classify the blobs by applying *cNet* on a subset of pixels (lines 10-19): each pixel "votes" for a class, the majority decides the class of the whole blob, blobs classified as '*soil*' are discarded.

4 Unsupervised Dataset Summarization

The CNNs described above should be trained using pixel-wise labeled datasets: unfortunately, pixel-wise data annotation is an extremely time consuming process, even if the user can exploit specific labeling tools that allow to quickly detect pixels belonging to vegetation by means of local thresholding operations based on NDVI. A first solution to this problem would be to extract and label only a subset of K images, randomly selected between the N images of the original dataset, $K \ll N$. Experimental evidence (Sec. 5) indicates that a randomly selected subset often does not well describe the original dataset, i.e. the subset provides a poor information "coverage" of the original dataset. Alternatively, the subset selection process could be done manually, by looking for a "good" subset of the sample images that well represent the original dataset: this strategy usually enables to obtain better classification results compared with randomly select subsets. We introduce here a simple but effective algorithm that enables to automatically select a subset of the training set that shows very good coverage properties over the original dataset. We call this problem *unsupervised dataset summarization*, where unsupervised means that the subset is extracted before the labeling process and summarization means that the subset must be very informative about the original dataset. This problem can be formulated as a special case of the *Knapsack Problem*, that given a set $\mathbf{V}$ of N elements, each one with a given weight c_i, asks for the subset $\mathbf{S}^*$ that maximize a *set* function $\mathcal{F} : 2^{\mathbf{V}} \rightarrow \mathbb{R}$ subject to a constraint that requires the total weight of the subset to be less or equal than a given threshold K:

$$\mathbf{S}^* = \underset{\mathbf{S} \subseteq \mathbf{V}}{\operatorname{argmax}} \, \mathcal{F}(\mathbf{S}) \; \textit{subject to} \; \sum_{i \in \mathbf{S}} c_i \leq K \tag{2}$$

The set function $\mathcal{F}$, also called *objective function*, measures the "quality" of a given subset. In our case the set $\mathbf{V}$ is the original dataset that contains N images, the constraint is represented by an equality constraint where for each i we have $c_i = 1$, while the set function $\mathcal{F}$ should tell us how well the subset $\mathbf{S}$ summarizes the original dataset $\mathbf{V}$. It is well know that this class of problems is NP-hard, so

the computation of the optimal solution $\mathbf{S}^*$ is often not feasible. Despite that, a good approximated solution can be obtained if we provide a objective function $\mathcal{F}$ that is *monotone submodular*. A set function $\mathcal{F}$ is submodular if for each $\mathbf{A} \subseteq \mathbf{B} \subseteq \mathbf{V}$ and for some element $x \notin \mathbf{B}$, we have that:

$$\mathcal{F}(\mathbf{A} \cup x) - \mathcal{F}(\mathbf{A}) \geq \mathcal{F}(\mathbf{B} \cup x) - \mathcal{F}(\mathbf{B}) \tag{3}$$

A submodular set function is monotone if for each $\mathbf{A} \subseteq \mathbf{B}$ we have $\mathcal{F}(\mathbf{A}) \leq \mathcal{F}(\mathbf{B})$. Submodular functions have a very attractive property [16]: it can be proven that if $\mathcal{F}$ is monotone submodular, then $\mathcal{F}(\hat{\mathbf{S}}) \geq \left(1 - \frac{1}{e}\right) \mathcal{F}(\mathbf{S}^*) \approx 0.632\ \mathcal{F}(\mathbf{S}^*)$[2], with $\hat{\mathbf{S}}$ an approximated solution computed using a greedy algorithm.

4.1 Subset Selection as a Document Summarization

Our method is inspired by the document summarization task that, given a set $\mathbf{V}$ that contains all the sentences of a text document, searches for a subset of sentences $\mathbf{S} \subseteq \mathbf{V}$ that well represents the original document. Typically this task is subject to some constraints, such as the maximum number of words or the maximum number of sentences that compose the subset.
Let us consider a dataset acquired by a robot moving in the field as the original "document" $\mathbf{V}$, possibly composed by thousands of images. If we consider each image as a "sentence" of $\mathbf{V}$, each one composed by a set of "visual words" [23], we can reduce our problem of subset selection as a standard document summarization problem. Lin and Bilmes [14] faced the document summarization problem by proposing a class of submodular set functions that measure both the similarity of the subset $\mathbf{S}$ to the document to be summarized (also called "coverage" of the original document) and the "diversity" of the sentences that compose the subset $\mathbf{S}$. Since our goal is to encourage subset $\mathbf{S}$ that well describe $\mathbf{V}$, we employ as objective function a simple but in our case effective coverage set function:

$$\mathcal{L}(\mathbf{S}) = \sum_{i \in \mathbf{V}, j \in \mathbf{S}} w_{ij} \tag{4}$$

where $w_{ij} \geq 0$ represents a similarity between the image (i.e., "sentence") i and the image j. $\mathcal{L}(\mathbf{S})$ is clearly monotone submodular.

4.2 Bag-of-Visual-Words from the CNN

In the document summarization task sentences are usually represented using bag-of-terms vectors: in a similar way, we represent each image using bag-of-visual-words vectors [23]. Since the goal is to train a CNN (in our case, the CNN of Fig. 3) using a very informative subset of the original dataset, we would like to extract the visual words *directly* from the trained CNN. In a typical

[2] This is a lower bound: in most of the practical cases the approximated solution ensures much better results.

CNN architecture, the sequence of convolutional layers usually computes a n-dimensional vector f, used as input of a sequence of fully connected layers: the decision over the output classes depends only on f. Such a vector represents a descriptor, or *signature*, of the input image or patch. In our specific case, we apply the *cNet* of Fig. 3 to 61×61 possibly overlapping patches of the input image. After two convolutional + pooling layers (blue dotted box in Fig. 3) the patch is reduced to a 384-dimensional vector f. The idea is to represent an image as a collection of m visual words, derived from the vectors $f_i,\ i = 1, \ldots, m$ provided by the CNN applied to m patches. If we denote with W the cardinality of our vocabulary trough visual words, we can quantize the descriptors f into visual words exploiting the k-means clustering algorithm [3]. The bag-of-visual-words vector for a given image is simply the W-dimensional histogram that reports the number of times that each visual word α appears in the image.
We computed w_{ij} using the following cosine similarity:

$$w_{ij} = \frac{\sum_{\alpha \in \mathbf{S}_i} (h_{\alpha,i} \cdot h_{\alpha,j} \cdot ih_{\alpha}^2)}{\sqrt{\sum_{\alpha \in \mathbf{S}_i} (h_{\alpha,i}^2 \cdot ih_{\alpha}^2)} \sqrt{\sum_{\alpha \in \mathbf{S}_j} (h_{\alpha,j}^2 \cdot ih_{\alpha}^2)}} \tag{5}$$

where $h_{\alpha,i}$ and $h_{\alpha,i}$ are the number of times that the visual word α appears in the image, and ih_{α} is the inverse document frequency, that is calculated as the logarithm of the ratio of the number of images where α appears, over the total number of images N that compose the input dataset.

4.3 The Proposed Algorithm

The proposed method is not directly applicable: we tacitly assumed that the CNN is already able to provide valid results even if we are still training it (i.e., we are looking for a good subset of the original data set to be used for training). We solve this issue by pre-training the CNN using a general labeled auxiliary dataset or a randomly selected, manually labeled subset of the input dataset.
The pseudo-code of our Unsupervised Dataset Summarization technique is reported in Algorithm 2: we first compute the CNN descriptors from a set of patches (lines 1-4), we then extract the bag-of-visual-words vectors (lines 5-8) and finally we select the subset $\mathbf{S}$ using a simple greedy algorithm that exploits the coverage set function reported in Eq. 4 and the similarity between images reported in Eq. 5 (lines 9-13).

5 Experimental results

The experimental results presented this section are designed to show the accuracy of our classification system. They also confirm the performances reached by a CNN trained on a small and very representative dataset, build up by our unsupervised dataset summarization approach.

Algorithm 2: Unsupervised Dataset Summarization

Data: The input dataset $\mathbf{V}$ with N images $\mathcal{I}$, the size W of the visual word vocabulary, the size K of the output subset
Result: The selected subset $\mathbf{S}$

foreach $\mathcal{I} \in \mathbf{V}$ **do**
  Extract in a fixed grid a number of m patches;
  For each patch, compute the descriptor f provided as output of the convolutional layers of the pre-trained CNN;
end foreach
Quantize all the descriptors into W visual words using the k-means algorithm;
foreach $\mathcal{I} \in \mathbf{V}$ **do**
  Compute the W-dimensional histogram that reports the numbers of times that each visual word α appears in $\mathcal{I}$;
end foreach
$\mathbf{S} \leftarrow \{\}$;
for $k \leftarrow 1$ **to** K **do**
  $\mathcal{I}^* \leftarrow \underset{\mathcal{I} \in \mathbf{V} \setminus \mathbf{S}}{\operatorname{argmax}} \mathcal{L}(\mathbf{S} \cup \{\mathcal{I}\})$;
  $\mathbf{S} \leftarrow \mathbf{S} \cup \{\mathcal{I}^*\}$;
end for

5.1 Experimental setup

We use two datasets, both collected from a BOSCH Bonirob farm robot (Fig. 1(a)) moving on a sugar beet field. Both the datasets are composed by a set of images taken by a 1296×966 pixels 4-channel JAI AD-130 camera mounted on the Bonirob. During the acquisition, the camera pointed downwards on the field and took images with a frequency of 1 Hz.
The first dataset (Dataset A) is composed by 700 images and it has been collected in the first growth stage of the plants, when both crop and weeds have not yet developed their complete morphological features. The second dataset (Dataset B) is composed by 900 images and it has been collected after 4 weeks: plants in this case are in an advanced growth stage. From each dataset we extract different subsets, each one manually labeled.
The performances of our classification approach have been measured by using two widely used metrics: the mean accuracy (MA, Eq. 6) and the mean average precision (MaP, Eq. 7):

$$MA = \frac{1}{N} \sum_{n=1}^{N} \frac{T_{pos} + T_{neg}}{T_{pos} + F_{pos} + T_{neg} + F_{neg}} \tag{6}$$

$$MaP = \frac{1}{Q} \sum_{q=1}^{Q} AP(q) \tag{7}$$

where T_{pos} and F_{pos} are the numbers of true and false positives, T_{neg} and F_{neg} are the numbers of true and false negatives, and $AP(q)$ is the average precision.

We implemented and trained the proposed CNNs *cNet* and *cNet* using the open source library TensorFlow [1].

5.2 Vegetation Detection

The first set of experiments is designed to show the performances of our vegetation detection approach that makes use of a conservative NDVI segmentation as initial pixel segmentation (see Sec. 3.1).
We train different networks in terms of amount and sizes of convolutional and fully connected layers by using the same training set taken from the dataset A. The results are shown in Table 1. We achieve the best mean average precision and accuracy (96.8% and 91.3%, respectively) with the biggest networks, composed by two convolutional and two fully connected layers. Nevertheless, our choice is to use the *sNet1c10-1f20*, being it a perfect trade-off between average time and accuracy. We compared the performance of this network with the standard NDVI based vegetation detection algorithm for some fixed thresholds (Table 2): the results of *sNet* are remarkable since it outperforms NDVI in all cases while it does not depend on any threshold.

Table 1: Vegetation detection results for different *sNet* networks. The network names follow the convention: $sNet < x > c < y > - < z > f < w >$, where x: number of convolutional layers, y: size of output feature maps, z: number of fully connected layers, w: size of the fully connected layers.

Net Type	MA	MaP	Average Time[s]
sNet1c10-1f20	96.7%	91.2%	0.43
sNet1c5-1f10	96.6%	90.8%	**0.34**
sNet1c20-1f40	96.7%	91.2%	0.45
sNet2c10-2f20	**96.8%**	**91.3%**	1.05
sNet2c5-2f10	**96.8%**	**91.3%**	0.98
sNet2c20-2f40	**96.8%**	**91.3%**	1.23

Table 2: Comparison between the NDVI threshold based vegetation detection and the *sNet1c10-1f20*

Net Type	*sNet*	$NDVI_{160}$	$NDVI_{170}$	$NDVI_{180}$	$NDVI_{190}$	$NDVI_{200}$
Mean Accuracy	**96.7%**	90.2%	95.6%	96.4%	95.2%	92.3%

5.3 Crop/Weed Classification

In order to show the classification accuracy of our pipeline, we perform experiments for both the pixel-wise and blob-wise approaches. The results of a comparison among different networks in the case of pixel-wise classification are reported in Table 3(a). As described in Sec. 3, we use a combination of a *sNet* followed by a *cNet*. We report the average timing results for sample steps of 1 (i.e., the *cNet* is applied to each active pixel) and 3 pixels (i.e., the *cNet* is applied on a grid with spacing 3 by 3 pixels). The best trade-off in terms of accuracy, precision and computational time is obtained by the combination *sNet1c10-1f20 + cNet2c64-2f192* , where the *cNet* is composed by four layers, equally divided into two convolutional and two fully connected layers. This network reaches a MaP of 96.1% with a lower computational time with respect to the others. We also compare the combinations of *sNet* and *cNet* with the *cNet* network used alone. In this case the *cNet* has to be applied to the whole image, and the complete image classification is done in 23 seconds without any significant increase in precision. Examples of pixel-wise and grid classification are shown in Fig. 4(a) and 4(b).
In Table 3(b) we report the classification performances obtained using our blob-wise classification algorithm (Seq. 4.3). The results are remarkable since the reported statistics refer only to the image pixels classified as vegetation by the *sNet*. We obtain these results without employing any plant position prior. Moreover, the timing results meet the real-time constraints required by our system. Some qualitative results are reported in Fig. 4(c).

5.4 Unsupervised Dataset Summarization

We finally evaluated the performances of our unsupervised dataset summarization algorithm. We compared the pixel-wise classification results using a CNN similar to the *cNet* depicted in Fig. 3, with a descriptor size of 384 entries (i.e. the size of the first fully connected layer). We used subsets of $K = 50$ images for both dataset A and dataset B: we trained the CNN using K randomly chosen images taken only from the dataset A (*cNetRandomA* in Table 4) and K randomly chosen images taken only from the dataset B (*cNetRandomB* in Table 4). We repeated the training steps using K images manually chosen from the dataset A (*cNetManualA* in Table 4) and K images manually chosen from the dataset B (*cNetManualB* in Table 4): in both cases, we looked for subsets that well represent the original dataset. We finally trained the CNN using the automatically selected subsets obtained by applying Algorithm 2 (*cNetUdsA* and *cNetUdsB*), where we used *cNetRandomA* and *cNetRandomB* as pre-trained CNNs, respectively, and a vocabulary of $W = 4096$ visual words. We also performed a cross-validation of the trained CNNs, evaluating a CNN trained with the dataset A with a validation set extracted from dataset B, and vice versa. As shown in Table 4, the network trained by using subsets selected by our unsupervised dataset summarization algorithm outperform in all the evaluations the network trained with the manually and the randomly chosen training sets,

Table 3: Classification results for different *cNet* networks. The network names follow the convention: $cNet < x > c < y > - < z > f < w >$, where x: number of convolutional layers, y: size of output feature maps, z: number of fully connected layers, w: size of the fully connected layers.

(a) Pixel-wise and 3×3 grid classification

Net Type	MA	$MA_{3\times3}$	MaP	$Map_{3\times3}$	Time[s]	$Time_{3\times3}$[s]
scNet2c64-2f192	92.3%	93.3%	96.2%	95.6%	200	23
sNet1c10-1f20 + cNet2c64-2f192	91.7%	91.8%	96.1%	94.3%	25-37	2.8-3.4
sNet1c10-1f20 + cNet2c32-2f100	90.8%	90.7%	95.2%	94.1%	**22-35**	**2.5-3.2**
sNet1c10-1f20 + cNet2c96-2f384	91.7%	91.7%	97.2%	94.5%	28-40	2.6-3.3
sNet1c10-1f20 + cNet3c64-3f192	91.8%	91.8%	**97.4%**	**95.7%**	33-48	3.6-4.5
sNet1c10-1f20 + cNet3c96-3f384	**92%**	**91.9%**	**97.4%**	94.9%	35-50	3.7-4.9

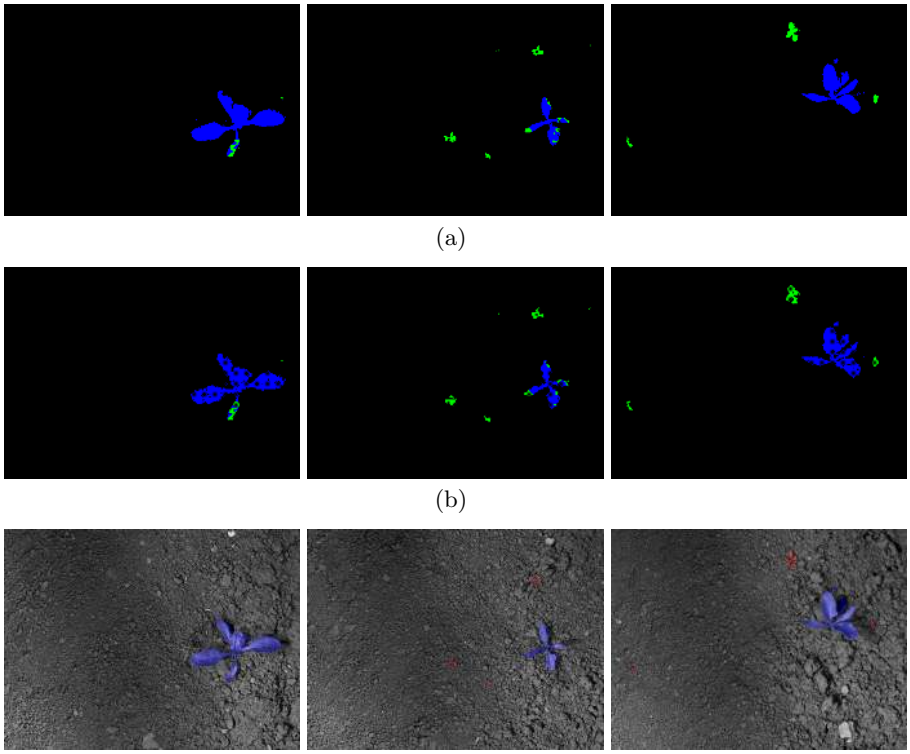
(b) Blob-wise classification

Net Type	MA	MaP	Time[s]
scNet2c64-2f192	92.3%	96.2%	23
sNet1c10-1f20 + cNet2c64-2f192	97.1%	98.3%	0.99
sNet1c10-1f20 + cNet2c32-2f100	95.6%	97.8%	**0.93**
sNet1c10-1f20 + cNet2c96-2f384	97.2%	98.3%	1.02
sNet1c10-1f20 + cNet3c64-3f192	**98%**	98.3%	1.74
sNet1c10-1f20 + cNet3c96-3f384	**98%**	**98.7%**	2.01

in both datasets A and B. The relatively poor classification results (59.4%) obtained with a CNN tested with a subset of the dataset B and trained using samples taken from dataset A are due to the fact that the dataset A includes only plants that are in their first growth stage, thus without their complete morphological features.

Table 4: Pixel-wise classification performances comparison for both datasets A and B for a *cNet* trained with different trainings sets.

TrainSet & Dataset	MaP A	MaP B
cNetRandomA	94.5%	57.7%
cNetManualA	95.4%	57.9%
cNetUdsA	**96.1%**	**59.4%**
cNetRandomB	78.1%	97.5%
cNetManualB	79.1%	98.6%
cNetUdsB	**82.3%**	**99.4%**



(a)

(b)

(c)

Fig. 4: (a)(b) Pixel-wise and 3x3 grid based classification mask outputs from the sNet1cm1fm + cNet2cm2fm network: in black, green and blue are represented, respectively, pixels that belong to soil, weed and crop; (c) Final blob-wise classification outputs from sNet1cm1fm + cNet2cm2fm network: pixels that belong to crop are highlighted in violet, pixel that belong to weeds are highlighted in red.

Globally, our results are comparable with the ones recently reported in [15], obtained using the same datasets but, differently from [15], we *do not* exploit any row arrangement. We expect to obtain even better results by integrating also this type of information.

6 Conclusions

In this work we addressed the problem of plant detection and crop/weed classification trough a multi-spectral camera mounted on a ground robot. We leverage the effectiveness of the convolutional neural networks by proposing the following contributions: (a) A parameterless vegetation detection approach that outperforms conventional methods based on the Normalized Difference Vegetation

Index (NDVI); (b) A fast classification pipeline that achieves state-of-the-art results by exploiting a sequence of a lightweight CNN followed by a deeper CNN that votes on connected vegetation blobs; (c) A dataset summarization algorithms that enables to streamline and speed-up the manual dataset labeling process while preserving good classification performances. The latter represents the main contribution of this paper. We reported detailed validations of each contribution, where we used real datasets taken from a farm robot moving in a sugar beet field. The results confirm the effectiveness of the proposed solutions.

Acknowledgement. We thank Cyrill Stachniss and Philipp Lottes for providing us with the datasets used in this paper.
This work has been supported by the European Commission under the grant number H2020-ICT-644227-FLOURISH.

References

1. Abadi, M., *et al.*: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), http://tensorflow.org/
2. Aitkenhead, M., Dalgetty, I., Mullins, C., McDonald, A., Strachan, N.: Weed and crop discrimination using image analysis and artificial intelligence methods. Computers and Electronics in Agriculture 39(3), 157–171 (2003)
3. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc. (2006)
4. Borregaard, T., Nielsen, H., Nrgaard, L., Have, H.: Cropweed discrimination by line imaging spectroscopy. Journal of Agricultural Engineering Research 75(4), 389–400 (2000)
5. Burks, T.F., Shearer, S.A., Gates, R.S., Donohue, K.D.: Backpropagation neural network design and evaluation for classifying weed species using color image texture. Transactions of the ASAE 43(4), 1029–1037 (2000)
6. Cerutti, G., Tougne, L., Mille, J., Vacavant, A., Coquin, D.: A model-based approach for compound leaves understanding and identification. In: IEEE International Conference on Image Processing. pp. 1471–1475 (2013)
7. Che Hussin, N., Jamil, N., Nordin, S., Awang, K.: Plant species identification by using scale invariant feature transform (SIFT) and grid based colour moment (GBCM). In: Proc. of the IEEE Conference on Open Systems (ICOS). pp. 226–230 (2013)
8. Feyaerts, F., Gool, L.V.: Multi-spectral vision system for weed detection. Pattern Recognition Letters 22(6-7), 667–674 (2001)
9. Haug, S., Michaels, A., Biber, P., Ostermann, J.: Plant classification system for crop /weed discrimination without segmentation. In: Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2014)
10. Hemming, J., Rath, T.: PA-precision agriculture: Computer-vision-based weed identification under field conditions using controlled lighting. Journal of Agricultural Engineering Research 78(3), 233–243 (2001)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proc. of a Advances in Neural Information Processing Systems (NIPS). pp. 1106–1114 (2012)

12. Kumar, N., Belhumeur, P.N., Biswas, A., Jacobs, D.W., Kress, W.J., Lopez, I., Soares, J.V.B.: Leafsnap: A computer vision system for automatic plant species identification. In: The 12th European Conference on Computer Vision (ECCV). pp. 502–516 (2012)
13. Lee, S.H., Chan, C.S., Wilkin, P., Remagnino, P.: Deep-plant: Plant identification with convolutional neural networks. In: Proc. of the IEEE International Conference on Image Processing (ICIP). pp. 452–456 (2015)
14. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. vol. 37, pp. 510–520 (2011)
15. Lottes, P., Hoeferlin, M., Sander, S., Müter, M., Schulze Lammers, P., Stachniss, C.: An effective classification system for separating sugar beets and weeds for precision farming applications. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA) (2016)
16. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. Mathematical Programming 14(1), 265–294 (1978)
17. Parkhi, O.M., Vedaldi, A., Jawahar, C.V., Zisserman, A.: Cats and dogs. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 3498–3505 (2012)
18. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8), 1226–1238 (2005)
19. Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning deep convolutional networks for plant recognition. In: Working Notes of Conference and Labs of the Evaluation forum (CLEF) (2015)
20. Rouse, Jr., J.W., Haas, R.H., Schell, J.A., Deering, D.W.: Monitoring vegetation systems in the great plains with ERTS. In: Proc. of the 3rd Earth Resource Technology Satellite (ERTS) Symposium. vol. 1 (1974)
21. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2014)
22. Shearer, S.A., Holmes, R.G.: Plant identification using color co-occurrence matrices. Transactions of the ASAE 33(6), 2037–2044 (1990)
23. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: Proc. of the IEEE International Conference on Computer Vision (ICCV) (2005)
24. Tellaeche, A., Burgos-Artizzu, X.P., Pajares, G., Ribeiro, A.: A vision-based method for weeds identification through the bayesian decision theory. Pattern Recognition 41(2), 521–530 (2008)
25. Vinh, L.T., Lee, S., Park, Y., d'Auriol, B.J.: A novel feature selection method based on normalized mutual information. Applied Intelligence 37(1), 100–120 (2011)
26. Wang, X.F., shuang Huang, D., xiang Du, J., Xu, H., Heutte, L.: Classification of plant leaf images with complicated background. Applied Mathematics and Computation 205(2), 916–926 (2008)
27. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 1577–1584 (2011)