Computers and Electronics in Agriculture 143 (2017) 314–324

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Weed detection in soybean crops using ConvNets

Alessandro dos Santos Ferreira[a,*], Daniel Matte Freitas[a], Gercina Gonçalves da Silva[a], Hemerson Pistori[b], Marcelo Theophilo Folhes[c]

[a] Federal University of Mato Grosso do Sul, Brazil
[b] Dom Bosco Catholic University, Brazil
[c] IbiGeo Geociência Aplicada, Brazil

ARTICLE INFO

Keywords:
Deep Learning
Weed detection
Computer vision

ABSTRACT

Weeds are undesirable plants that grow in agricultural crops, such as soybean crops, competing for elements such as sunlight and water, causing losses to crop yields. The objective of this work was to use Convolutional Neural Networks (ConvNets or CNNs) to perform weed detection in soybean crop images and classify these weeds among grass and broadleaf, aiming to apply the specific herbicide to weed detected. For this purpose, a soybean plantation was carried out in Campo Grande, Mato Grosso do Sul, Brazil, and the Phantom DJI 3 Professional drone was used to capture a large number of crop images. With these photographs, an image database was created containing over fifteen thousand images of the soil, soybean, broadleaf and grass weeds. The Convolutional Neural Networks used in this work represent a Deep Learning architecture that has achieved remarkable success in image recognition. For the training of Neural Network the CaffeNet architecture was used. Available in Caffe software, it consists of a replication of the well known AlexNet, network which won the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). A software was also developed, Pynovisão, which through the use of the superpixel segmentation algorithm SLIC, was used to build a robust image dataset and classify images using the model trained by Caffe software. In order to compare the results of ConvNets, Support Vector Machines, AdaBoost and Random Forests were used in conjunction with a collection of shape, color and texture feature extraction techniques. As a result, this work achieved above 98% accuracy using ConvNets in the detection of broadleaf and grass weeds in relation to soil and soybean, with an accuracy average between all images above 99%.

1. Introduction

Soybean represents the main oilseed consumed worldwide, both for animal consumption through soybean meal and for human consumption through oil (Silva et al., 2011). According to data from 2016, more than 340 million tonnes of soybeans were produced worldwide (International: World soybean production, 2017). Given the importance of soy in the economic context, it is essential to use techniques to maximize productivity and product quality.

Weeds are plants that grow spontaneously on agricultural soils where they are unwanted. The growth of these plants competing with economic crops, such as soybeans, causes damages, making it difficult to operate harvesting machines and increasing the impurity and moisture of the grains (Voll et al., 2005). The negative effects of weeds on crops include competition to water, light, nutrients and space, increased production costs, difficulty in harvesting, depreciation of product quality, increase risk of pests and diseases and decrease in the commercial value of cultivated areas (Rizzardi and Fleck, 2004). Weeds can be classified based on the format of their leaves into grass and broadleaf categories. The separation in these two classes is adequate because grass and broadleaf weeds are differentiated in the treatment due to the selectivity of some herbicides to the specific group (Herrera et al., 2014). Herbicide application achieves better results if the treatment is targeted to the specific class of weeds.

The knowledge of weed infestation is a fundamental procedure for the use of preventive measures in their control (Rizzardi and Fleck, 2004). It is important to have methods that quantify and analyze the distribution of weed infestation quickly and economically, and a more practical method is required than the systematic observation of crops. Several studies have been carried out recently aiming automation of this process of identification and classification of weeds. Ahmed et al. (2012) used Support Vector Machines (SVMs) to identify six weed

* Corresponding author at: Federal University of Mato Grosso do Sul, 368 Brigadeiro Machado Street Block 4 Apt. 112, Campo Grande, MS 79006-610, Brazil.
E-mail addresses: asf2005kn@hotmail.com (A. dos Santos Ferreira), daniel.freitas@ufms.br (D. Matte Freitas), Gercina.goncalves@gmail.com (G. Gonçalves da Silva), pistori@ucdb.br (H. Pistori), marcelo@ibigeo.com.br (M. Theophilo Folhes).

https://doi.org/10.1016/j.compag.2017.10.027
Received 8 February 2017; Received in revised form 26 July 2017; Accepted 28 October 2017
Available online 12 November 2017
0168-1699/

species in a database of 224 images achieving 97.3% accuracy with the best combination of extractors. Hung et al. (2014) classified three weed species using Sparse Autoencoders with precision of 72.2%, 92.9% and 94.3% for each specie. Cheng and Matson (2015) used multiple machine learning algorithms such as Decision Tree, SVM and Neural Network in rice and weed discrimination on images downloaded from internet. They achieved 98.2% precision as the best result. There are several recent studies concerning the classification between grasses and broadleaf weeds. Herrera et al. (2014) using shape descriptors and Fuzzy Decision-Making achieved 92.9% accuracy in a set of 66 images. Siddiqi et al. (2014) used SVMs and achieved 98.1% accuracy with 1200 images. Ishak et al. (2009) reached 93.75% in the identification of grasses and broadleaf weeds in a dataset of 400 images using Single Layer Perceptron. Other studies using SVMs have been carried out in the identification of weeds (Tellaeche et al., 2011; Saha et al., 2016).

In this work the objective is to use Deep Convolutional Neural Networks or ConvNets to perform the identification of weeds in relation to soybean and soil and classification of them in grass and broadleaf. Deep Learning, in particular the architecture of Convolutional Neural Networks, represent currently the state of the art in the recognition of images and objects (LeCun et al., 2015). Classifiers used for this purpose as Support Vector Machines and traditional Neural Networks (MLPs) are dependent on good feature extractors such as Scale-Invariant Feature Transform (SIFT) and Gabor wavelet (Siddiqi et al., 2014). The key factor of Deep Learning is that its feature extraction is learned automatically from the raw data (LeCun et al., 2015). Its attributes layers are not modeled manually and it is not necessary to use extractor algorithms to perform the training and classification of the Convolutional Network. Thus, it is expected that this technique will be much more successful in the near future because it requires little engineering by hand and may benefit from increased computing capacity and data (LeCun et al., 2015).

Convolutional Neural Networks were inspired by the structure of the visual system. The first computational models based on these local connectivities between neurons and on hierarchically organized image transformations are found in Fukushima's Neocognitron (Fukushima, 1980). Later LeCun, using the Convolutional Networks architecture, achieved the state of the art in various image recognition tasks (LeCun et al., 1990). The modern understanding of the physiology of the visual system is consistent with the processing style found in the convolutional networks (Bengio, 2009). In some cases, Gabor filters have been used as an initial preprocessing to emulate the human visual response to visual perceptions (Arel et al., 2010). In addition to the use of image recognition, variations of Convolutional Neural Networks have been obtaining excellent results in other areas such as natural language processing (Kalchbrenner et al., 2014; Dos Santos and Gatti, 2014), prediction of the bioactivity of molecules for drug discovery (Wallach et al., 2015) and generation of captions from images (LeCun et al., 2015).

This work presents a broader approach in relation to the studies cited. In addition to classifying images into correspondent class, such as some of the studies mentioned, the Pynovisão software was built. This software, through the Simple Linear Iterative Clustering (SLIC) Superpixels algorithm, allowed the detection of weeds in crop images captured by Unmanned Aerial Vehicles (UAVs), returning a visual classification of the presence of weeds in the photographed area. The Pynovisão also allowed the creation of a robust image dataset, containing more than fifteen thousand images, a value significantly higher than the works cited. Finally, this study performs the classification using ConvNets, an architecture that has achieved excellent results in image recognition, and compares the results with classifiers that have been traditionally adopted successfully in the problem of detection and discrimination of weeds, as Support Vector Machines.

2. Materials and methods

2.1. Overview

The approach proposed in this work for the detection of weeds is composed of five stages. The first stage consists of the capture of images of soybean crops, for which we use UAVs. The second stage is the segmentation of these images using the algorithm of superpixels, whose extracted segments were annotated manually. These segments are used in the construction of an image dataset of soil, soybean and weeds. In the third stage, exclusive to the classifiers that will be used in the comparison to the ConvNets performance, we performed the feature extraction of the image dataset segments with a collection of extractors of color, shape and texture. The fourth stage consists of the training of classifiers. ConvNets perform this training using the image dataset segments as the other classifiers use the feature matrix obtained in the third stage of this process. The last phase consists of the segmentation and classification of the image of a soybean plantation, returning visual classification regarding the presence of weeds in the image.

2.2. Image acquisition

The soybean plantation was carried out at the São José farm (Fig. 1), Campo Grande, Brazil, located under the geographical coordinates of latitude 20°24′9.88″S and longitude 54°36′31.49″W, in an area of one hectare. Flights were conducted between the months of December 2015 and March 2016 at least once a week, between eight and ten in the morning.

Fig. 1. Soybean plantation in São José farm, Campo Grande, Brazil.

Fig. 2. UAV DJI Phantom 3 Professional used for image acquisition.

A quadrocopter platform with vertical take-off and landing, model DJI Phantom 3 Professional was used to collect a set of aerial images (see Fig. 2). This model is equipped with a visible light RGB camera mounted on a 3-axis gimbal to ensure that images are acquired with the camera's optical axis pointing in the nadir direction, was flown over the test field on a weekly basis. The RGB camera was a Sony EXMOR 1/2.3″ with a 4000 × 3000 pixel detector used in most other consumer-grade drones. The UAV weighs approximately 1280 g with payload, and has an operational flight time of 20 min.

The images were captured in a manual mode at an average altitude of 4 m above ground level, which corresponds to a vertical photography ground sampling distance of less than 1 cm, covering all hectare test field, with a small variation in acquisition altitudes due the flight in the manual mode. The ratio of the image used was 4 × 3, with a resolution of 4000 × 3000 pixel and the image file format was JPG. The acquisition was performed with all the parameters in the default factory configuration. No additional image correction was applied.

2.3. Segmentation

2.3.1. SLIC superpixels

Superpixel algorithms group the pixels into atomic regions that can be used as substitutes for the pixel grid. They capture redundancy in the image by generating a structure that significantly reduces the complexity of image processing tasks. These algorithms have demonstrated great utility in applications such as object localization and image segmentation. We use the algorithm Simple Linear Iterative Clustering (SLIC) Superpixels (Achanta et al., 2012) to segment the images and assist in the construction of an image dataset.

The strategy of this algorithm consists in grouping pixels based on color similarity and spatial proximity in the image. For this we use the space of five dimensions $[labxy]$, where $[lab]$ is the color of the pixel in the CIELab color space and $[xy]$ represents the position of the pixel in the image. The algorithm receives as input the number of superpixels, of approximately same size, K. Thus, for an image with N pixels the approximate size of each superpixel is N/K pixels. In the case of superpixels of the same size, there would be a superpixel center in each grid in the range $S = \sqrt{N/K}$.

The SLIC algorithm constructs pixel clusters using a variation of the k-means algorithm that performs the search in a reduced space, proportional to the region of the superpixel, as opposed to the traditional algorithm that performs the comparison with all centers of the clusters. This behavior significantly increases the efficiency of the algorithm.

For this purpose, the algorithm defines K superpixel centers $C_k = [l_k,a_k,b_k,x_k,y_k]$, with $k = [1,K]$ in each grid in the range S. After this step, for each image pixel $P_i = [l_i,a_i,b_i,x_i,y_i]$ is calculated the proximity to the centers C_k using the distance defined in (1):

$$d_{lab} = \sqrt{(l_k-l_i)^2 + (a_k-a_i)^2 + (b_k-b_i)^2}$$
$$d_{xy} = \sqrt{(x_k-x_i)^2 + (y_k-y_i)^2}$$
$$D_s = d_{lab} + \frac{m}{S}d_{xy} \tag{1}$$

where D_s represents the sum of the distance in CIELab space, represented by variables lab, with the distance xy normalized in the plane by the interval S. The m variable represents the compactness of the superpixel and has the default value 10, defined in the original work.

2.3.2. SLIC parameters

By default, the only input parameter of the SLIC Superpixel algorithm is the number of superpixels, of approximately the same size, K. However, it is optionally possible to adjust the compactness parameter that allows controlling the shape of the superpixel making it more square. In the implementation used in this work, available in the library scikit-image (Van der Walt et al., 2014), it is also possible to configure the sigma parameter, which allows smoothing the image, using Gaussian filters, prior to segmentation.

Several tests were performed to define the value of K. In this problem the goal is not to segment the image at the leaf level, but to separate the image into segments that contain multiple leaves of soy or weeds. Segmenting the image with a high value for K results in smaller-sized superpixels, thus segmenting the image into well-defined classes as can be seen in Fig. 3. However, superpixels with small dimensions tend to store few descriptive characteristics of each class, impacting the quality of the classification. In order to define the value of K, multiple values of 100 were tested in the range of 100–1200.

The value $K = 300$ was chosen for dealing satisfactorily with both problems. The dimensions of a superpixel with these settings, for the images of dimensions 4000 × 3000 pixel used in this work, are approximately 200 × 200 pixel. These dimensions allowed simultaneously that each segment contained a relevant amount of information of the class that it represents besides a segmentation of the main image in well-defined classes. However, it is important to note that this parameter was chosen specifically for these images and are dependent on the height and resolution of image acquisition such as the segmentation objective.

For the compactness parameter, multiple values of 10 were tested in the range of 10–50. The objective of this evaluation was to find the value that best segmented the image, keeping in the same superpixel only elements belonging to the same class. In images that were captured on cloudy days with little shadow occurrence in the plantation, it was possible to use smaller values, leaving the superpixel borders more sensitive to edges of the image elements. However on sunny days, the effect of shadow on the image made the superpixels more sensitive to illumination of the image, including in the same superpixel segment elements belonging to different classes.

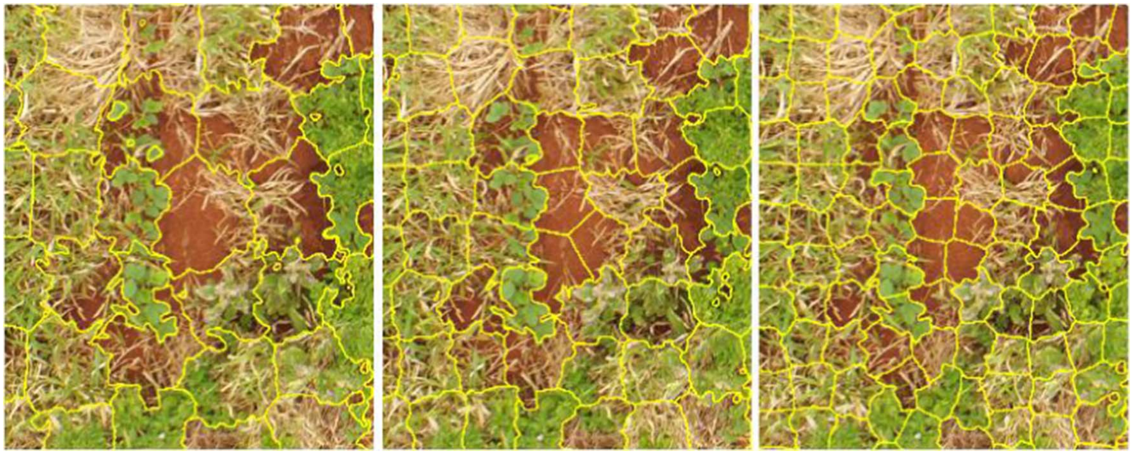

Fig. 3. From left to right, the image segmented with $K = 300,600$ and 1200. For the value 1200 the segments encompass the classes in a well defined way. However, the amount of relevant information in the segment is much lower than the segmentation with the value 300.

Fig. 4. On the left the segmented image with compactness 20, on the right with compactness 40. For the value 40 the segments are more sensitive to the variation of soybean and weeds in the image, grouping in the same superpixel, soybean leaves with different illumination rates. For the value 20 the segments are more sensitive to the variation of color caused by the effect of light and shadow.

In addition to this, the algorithm grouped into different superpixels adjacent elements of the same class due to the variation of light and shadow. The Fig. 4 illustrates this behavior. Thus, it was necessary to use higher values so that the information of spatial proximity had a greater weight in relation to the similarity of color and illumination. In order to standardize a value close to ideal for images with different types of illumination, a value of 40 for compactness has been chosen.

2.3.3. Image dataset

From the set of images captured by the UAV, all those with occurrence of weeds were selected resulting a total of 400 images. Through the Pynovisão software, using the SLIC algorithm, these images were segmented and the segments annotated manually with their respective class. These segments were used in the construction of the image dataset.

The segments of each image, that clearly identified one of the four classes used in this experiment, were annotated. Given the large presence of soybeans in the images, several segments of soybeans were arbitrarily ignored to avoid imbalance in the dataset.[1] Images from the month of January 2016 could not be used due to the poor quality caused by configuration issues with DJI Phantom 3 camera. These problems have been corrected only after resetting the camera to factory settings.

The image dataset was finalized with 15,336 segments, being 3249 of soil, 7376 of soybean, 3520 grass and 1191 of broadleaf weeds. In the Table 1 can be seen the division of images selected by date and the number of segments by classes.

[1] It is important to note that this decision introduces an extra artificiality in the image dataset.

Table 1
List of images containing weeds and the number of segments selected, by class. The images of the month of January of 2016 could not be used due low quality.

Date	Images	Soil	Soybean	Grass	Broadleaf	Segments
29/12/2015	22	541	53	152	509	1255
05/02/2016	69	875	1997	682	140	3694
16/02/2016	40	127	305	490	88	1010
18/02/2016	12	48	288	341	36	713
19/02/2016	38	315	935	373	4	1627
25/02/2016	40	11	82	267	36	396
26/02/2016	50	411	1791	128	158	2488
01/03/2016	29	3	6	221	8	238
04/03/2016	100	918	1919	866	212	3915
Total	400	3249	7376	3520	1191	15,336

2.4. Feature extraction

For the classifiers that we compared to ConvNets, after the segmentation step, we performed the feature extraction from each segment of the dataset using a collection of shape, color, texture and image orientation extractors implemented in the OpenCV (Bradski, 2000) and scikit-image (Van der Walt et al., 2014) libraries. The collection of extractors was composed of 218 features. A succinct description of each extractor used follows in the following sub-sections.

2.4.1. Gray-level co-occurrence matrix

A Gray-Level Co-occurrence Matrix (GLCM) stores information about the texture of an image. This information is stored in an array of frequencies P_{ij} with two neighboring pixels separated by a distance d in the image, one with gray level i and the other with gray tone j. These frequencies represent a function of the angular relationship and distance between neighboring pixels (Soh and Tsatsoulis, 1999).

We use in this work, in matrices 4×4 with distance 1 and 2 and angles 0°,45° and 90°, the following texture properties defined in the GLCM: energy, contrast, correlation, homogeneity and dissimilarity, resulting in 36 features.

2.4.2. Histogram of oriented gradients

Histogram of Oriented Gradients (HOG) is a feature extraction technique presented by Dalal and Triggs (2005), with the goal of detecting humans in images. The basic idea of the algorithm is that the shape and appearance of the object can be characterized by the distribution of local intensity gradients or edge directions. To achieve this, the image is divided into cells and for each cell a gradient histogram of each pixel contained in that cell is accumulated. To improve the invariance to illumination a contrast normalization is applied in blocks of cells overlapped on the image. The normalized blocks are defined as HOG descriptors.

For the Histogram of Oriented Gradients algorithm, 128 features were extracted and correspond to the 128 positions of the vector of HOGs calculated over the original image scaled to the dimensions 128×128 pixel.

2.4.3. Local binary patterns

Local Binary Patterns (LBP) are considered one of the best texture extractors being widely used in various applications (Ahonen et al., 2006). Its advantages are its invariance to changes in shades of gray and computational efficiency. Its strategy for detecting texture is to observe to a central point the variation of its color in relation to its neighbors. This procedure is performed for all pixels in the image being defined as center point and having its label assigned from the calculation with respect to its neighbors. The histogram of the pixel labels of the image is then used as a texture extractor.

2.4.4. Distribution of colors in RGB, HSV and CIELab spaces

We also used the minimum, maximum, mean and standard deviation attributes of each channel of RGB, HSV and CIELab color spaces.

2.5. Classifiers

In order to compare the performance of ConvNets, tests were performed with other classifiers. The algorithms used for the comparative tests were Adaboost, Random Forests and Support Vector Machines (SVM), using the implementation of Sequential Minimal Optimization (SMO) algorithm for the training. For the Adaboost M1 boosting algorithm, the chosen classifier was the C4.5 algorithm. For these algorithms was used the implementation available in the Weka software (Hall et al., 2009) and through the python-weka-wrapper library.

2.5.1. Adaboost – C4.5

Boosting is a technique used to improve the performance of a learning algorithm. In theory, boosting can be used to significantly improve any algorithm that generates classifiers whose results are little better than random guessing. AdaBoost is a boosting algorithm proposed by Freund and Schapire (1996). The Boosting technique works by running a weak learning algorithm over and over again using several distributions of the training set and then combining the classifiers obtained into a single aggregate classifier. However, although the technique focuses on improving the performance of weak classifiers, it can also be used with good classifiers such as C4.5 (Freund and Schapire, 1996).

C4.5 is an algorithm that uses decision tree to perform the classification of a set of test cases (Quinlan, 1993). A decision tree is a structure that consists of a leaf, which indicates a class contained in a test set, or a decision node, that specifies a test to be performed, generating a branch for each possible test result. This structure is used to perform the classification of an entry starting from the root of the tree and moving through it, performing the tests in each decision node, until a leaf is reached. In this work we use the J48 algorithm available in the Weka software, which consists of the Java re-implementation of the C4.5 Release 8 software, which contains additional features to the version described in Quinlan (1993).

2.5.2. Random Forests

Random Forests consist of a combination of decision trees formed from several random and independent samples of an input set and with the same distribution for all trees (Breiman, 2001). This technique is known as bagging, where several independently trees are grouped to provide a classification (Liaw and Wiener, 2002). The construction of these trees is performed by selecting a random sampling with replacement of the training set S. For each selected sample, a tree corresponding to that set is constructed.

The Random Forests algorithm also includes an additional layer of randomness to the traditional behavior of bagging. In addition to building each tree using a different sampling, the way each tree is constructed is also modified. In the traditional model each node is partitioned using the best partition among all the M attributes of the inputs. In this algorithm the choice of attributes is based on a subset of attributes of size m, where m is a fixed value for all constructed trees and $m < M$. These attributes are chosen at random and the construction of each tree is performed using only the subset of chosen attributes (Liaw and Wiener, 2002). The selection of the subset is also independent of all generated trees.

To perform the classification of a test case, the results provided by all trees for that input are grouped together. The class reported by Random Forests is defined from the majority of votes provided by all trees.

2.5.3. Support Vector Machines

Support Vector Machines (SVM) is a machine learning technique originally defined by Cortes and Vapnik (1995). In its simplest form, the linear form, support vector machines is a hyperplane such that there is a margin separating a set of positive and negative examples into a high dimensional space (Platt, 1998). Given the fact that there may be infinite choices for the margin separating these examples, the goal is to maximize the distance from that margin to the nearest negative and positive examples. In some cases it is not possible to linearly separate all cases from a set of data, so there is no hyperplane that divides all negative and positive points. In this case a penalty is applied to an example that fails to position itself at its correct margin. In addition, Support Vector Machines can be generalized to non-linear classifiers.

The training of Support Vector Machines requires the solution of the quadratic programming optimization problem. Platt (1998) proposed the Sequential Minimal Optimization (SMO) algorithm that presents a solution where it breaks the original quadratic programming problem into a series of smaller problems. This approach resulted in a decrease in training computing time and left the use of linear memory in the size of the training set, making it possible to train huge sets of tests efficiently. In this work we use the Sequential Minimal Optimization algorithm for the training of the Support Vector Machines.

2.5.4. ConvNets

For the training of ConvNets, the software Caffe (Jia et al., 2014) was used. Caffe is a framework for Deep Learning implemented in C++/CUDA for the training of Neural Networks. Its main motivation is image recognition, being widely used directly or as a basis for implementations of Neural Convolutional Networks. The topology of the Neural Network used was the CaffeNet network, a replication of the topology AlexNet (Krizhevsky et al., 2012).

AlexNet, used by Krizhevsky in ImageNet LSVRC 2012, is considered the boundary in the use of Neural Convolutional Networks for image recognition. It consists of 8 layers. The first 5 layers are convolutional and the last 3 layers are fully connected. The output of the last layer feeds a 1000-way sofmax, which produces the probabilistic distribution over the 1000 classes used in the competition. ReLUs

Fig. 5. Examples of image dataset classes. In the upper row examples of grass and broadleaf weeds classes are shown. In the lower row can be seen examples of soybean and soil classes.

(Rectified Linear Units) were used at the output of all convolutional and fully-connected layers (Krizhevsky et al., 2012). The original images provided for the training were supplemented by crops from original images and their horizontal reflections, to enlarge the training set.

2.6. Training and evaluation

All algorithms used for the comparative tests with ConvNets were executed with the default settings in Weka software version 3.6.6. Weka software consists of a collection of machine learning algorithms that have as input files the Attribute Relation Format File (ARFF) format, an ASCII text file that describes a list of instances, representing the input attribute matrix for the classifier. The ARFFs used as input in the Weka were generated from the extractors described in Section 2.4. The input ARFF file was created using the same dataset used by the ConvNets in their training and classification.

The images generated in the segmentation step were saved in .TIF format and with size proportional to the smallest rectangle that encompassed the entire superpixel selected (see Fig. 5). The CaffeNet network uses as input an image in the JPEG format and dimensions 256×256 pixel. To deal with this constraint, each .TIF image was placed into the upper left quadrant of an image composed of a black background, in JPEG format, with dimensions 512×512 pixel. After this step, the image was cropped, without resizing, in relation to the upper right quadrant with dimensions 256×256 pixel, to be submitted to training. Since most of the images of the dataset had height and width of less than 256 pixels, in less than 5% cases the crop caused loss of information.

To perform the evaluation tests, two sets were created from the image dataset. The first set was composed with balanced data, the images were divided so that all classes contained the same number of images.[1] The chosen number of images for each class was 1125, resulting in 4500 images. These 4500 images were automatically selected from the image dataset through a script that made the selection at random. Of these images, 3000 were used for the training, 500 for validation and 1000 for the tests. The validation images were not used in the training of the algorithms compared to the neural networks.

The second evaluated set was formed without the restriction that the classes were balanced. Without this restriction it was used 15,000 images of the 15,336 total images of the dataset. A multiple value of 1000 was chosen to facilitate splitting of the images into the training, validation and test sets. The 336 discarded images were selected in an analogous way to the method used in the first group, with the premise that all belong to the soybean class, because it has a higher number of images than the other classes, reducing the impact of the discard. For the other classes all the images of the dataset were used. This set was split with 70% of the images used for the training, 10% of the images used for validation and 20% of the images used for the final tests, following this ratio in all classes. As in the group of balanced data, the validation images were discarded for the algorithms compared to the neural networks.

In ConvNet training, for the dataset with balanced data, 7500 iterations were performed in the training set, where in each iteration a mini-batch of 50 images was used to train the network. The initial learning rate of the network was 10^{-3}, being multiplied by 10^{-1} every 3000 iterations. The final learning rate was 10^{-5}. For the set with unbalanced data, 15,000 iterations were performed, with the initial learning rate of 10^{-3}. However, in this set it was modified only once, with 10,000 iterations, to the value of 10^{-4}.

The final values of the parameters used in this training were obtained after empirical attempts of combinations. These empirical calculations were based on the evaluation of the accuracy of the network in the validation set and through the use of snapshots. Snapshots represents a copy of the network model in a given training iteration, which can be used to modify training parameters subsequent to that iteration. Using this structure, it was possible to test several parameter branches until the most consistent combination for the validation set was reached.

For the definition of the value of the initial learning rate, negative powers of 10 were evaluated in interval between 1 and 5, i.e., 10^{-i}, where $1 \leqslant i \leqslant 5$. Evaluations in the set of final tests were only performed after the definition of the final parameters, that is, the tests had no influence on the choice of parameters. Given the robustness of the model, no modification in the original architecture of the CaffeNet network was necessary. In addition to evaluations in the classification of the segments of the image dataset, the evaluation of the Pynovisão software and its performance in the detection and classification of weeds in soybean crop images captured by UAVs was made.

2.7. Pynovisão

The Pynovisão was a software developed in this work with the objective of being an integrated module of computer vision techniques. It performs tasks such as segmentation, feature extraction and image classification. For segmentation it provides three superpixel algorithms. SLIC was the chosen algorithm for this work. After dividing the images into superpixels it also allows the creation of an image dataset from the superpixels.

However the most important functionality of the Pynovisão for this work is the weed detection in a soybean plantation image. To achieve this objective, the software performs segmentation using the SLIC algorithm, through manually defined parameters. After the segmentation step, it saves all segments of the image and make the feature extraction in them. Finally, it performs the training of classifiers using images from the previously stored image dataset. With the trained classifier and the attribute matrix corresponding to the image segments, it performs the individual classification of each superpixel of the image and shows the visual result by painting each segment, as can be seen in Fig. 6.

For the ConvNets it is not necessary to perform the feature extraction, since the classification is done using the raw data of each image. However, the training of the neural network is not performed by the Pynovisão and it must be done by other software. For this work, the training and classification of ConvNets were carried out in an integrated way with Caffe software.

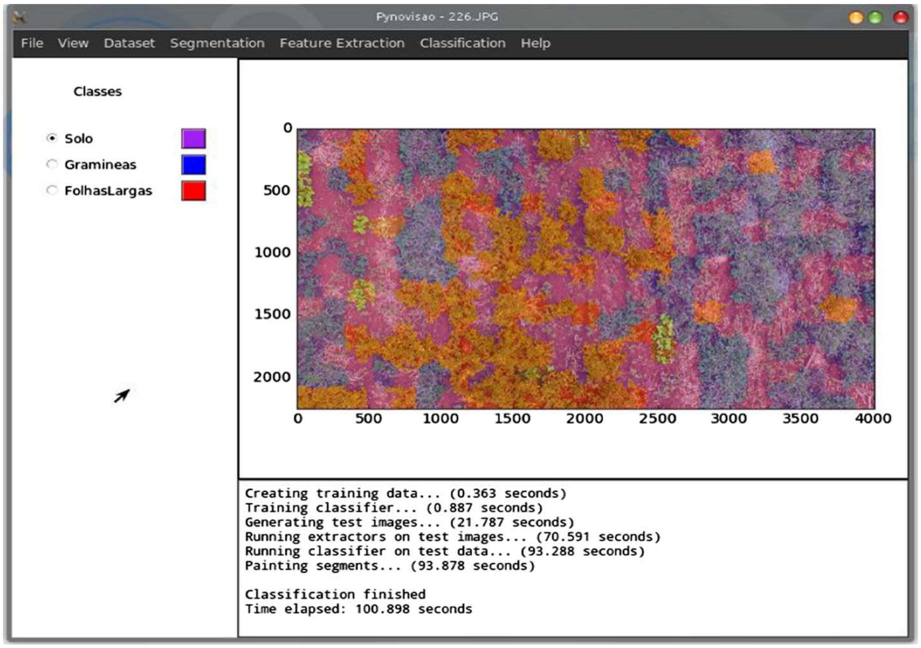

Fig. 6. Software Pynovisão.

3. Results and discussion

3.1. Evaluation with balanced data

For this evaluation, using 4500 images, the results reached high precision and sensitivity, as can be seen in the Table 2. These results demonstrate the efficiency of ConvNets in dealing with a problem containing a few classes when a robust training set is provided. It presented superior results to all other classifiers in all the metrics evaluated.

In the performance per class, the class soil had the best performance in all analyzed algorithms, since it presented completely different values of all other classes in the RGB and CIELab color spaces. In the identification of the weeds, the neural networks presented a superiority in the performance, in relation to the other algorithms, presenting values superior to 0.98 in the precision and sensitivity in the grasses and broadleaf weeds. However, it is important to note that, with the exception of Random Forests in the broadleaf precision, all the algorithms compared presented results above 90% in the accuracy and sensitivity of all classes presenting themselves as good alternatives to the problem.

Table 2
Confusion Matrix of all classifiers evaluated, for the group of 4500 images of the image dataset. The bold values represent the average precision and the average sensitivity of each classifier.

	Soil	Soybean	Grass	Broadleaf	Precision	Sensitivity
ConvNets						
Soil	250	0	0	0	1.000	1.000
Soybean	0	247	1	2	0.988	0.988
Grass	0	2	246	2	0.991	0.984
Broadleaf	0	1	1	248	0.984	0.992
Average					**0.991**	**0.991**
Support Vector Machines						
Soil	250	0	0	0	1.000	1.000
Soybean	0	241	3	6	0.953	0.964
Grass	0	9	237	4	0.967	0.948
Broadleaf	0	3	5	242	0.960	0.968
Average					**0.970**	**0.970**
Adaboost – C4.5						
Soil	249	0	1	0	0.996	0.996
Soybean	0	238	5	7	0.967	0.952
Grass	1	6	236	7	0.948	0.944
Broadleaf	0	2	7	241	0.945	0.964
Average					**0.964**	**0.964**
Random Forest						
Soil	246	3	0	1	0.996	0.984
Soybean	1	229	10	10	0.954	0.916
Grass	0	6	228	16	0.908	0.912
Broadleaf	0	2	13	235	0.897	0.940
Average					**0.938**	**0.938**

In Fig. 7 is possible to analyze the nine images classified incorrectly by the ConvNets. In the first broadleaf weeds image, it can be seen that there is also in the image the presence of grasses, which induced the classification in this wrong class. The same behavior is reproduced in the second image, where there are some soy leaves next to the weeds, causing the error in the identification. The first two images of soybeans correspond to the photographs captured in the month of December. These images were captured in the early stages of soybeans and were a little taller than the images obtained in 2016, resembling the images of broadleaf weeds captured in this period.

The third image on left, representing soybean, has stems that resemble the shape of grasses. However, it should be noted that this image was identified as grass with a probability lesser than 70%. Regarding the grasses images, it is notable that the great presence of shadow makes it difficult to identify the class in two images. However again it is interesting to note that eight of the nine incorrect classifications of the network were performed with a probability of less than 90%. This fact leads us to the analysis of probability classifications of ConvNets also in the images that were classified correctly, as can be seen in the Table 3.

It can be seen in the table that of the 1000 images classified by ConvNets, 480 were correctly classified with 100% probability. If we set a threshold of 0.98, we have 96.3% of the images were classified correctly and none of them received incorrect identification. The first image classified incorrectly appears only with the threshold of 0.96 and corresponds to broadleaf weeds that were classified as grasses with probability of 0.9738.

It is expected that in some images even trained specialists may make mistakes. These errors can be caused by poor image quality, shadowing influence or in case the image encompasses two or more classes in similar proportions. Using the probability information it is possible to adapt the classification to return positive identification only in cases where the probability exceeds a predefined threshold, which can be

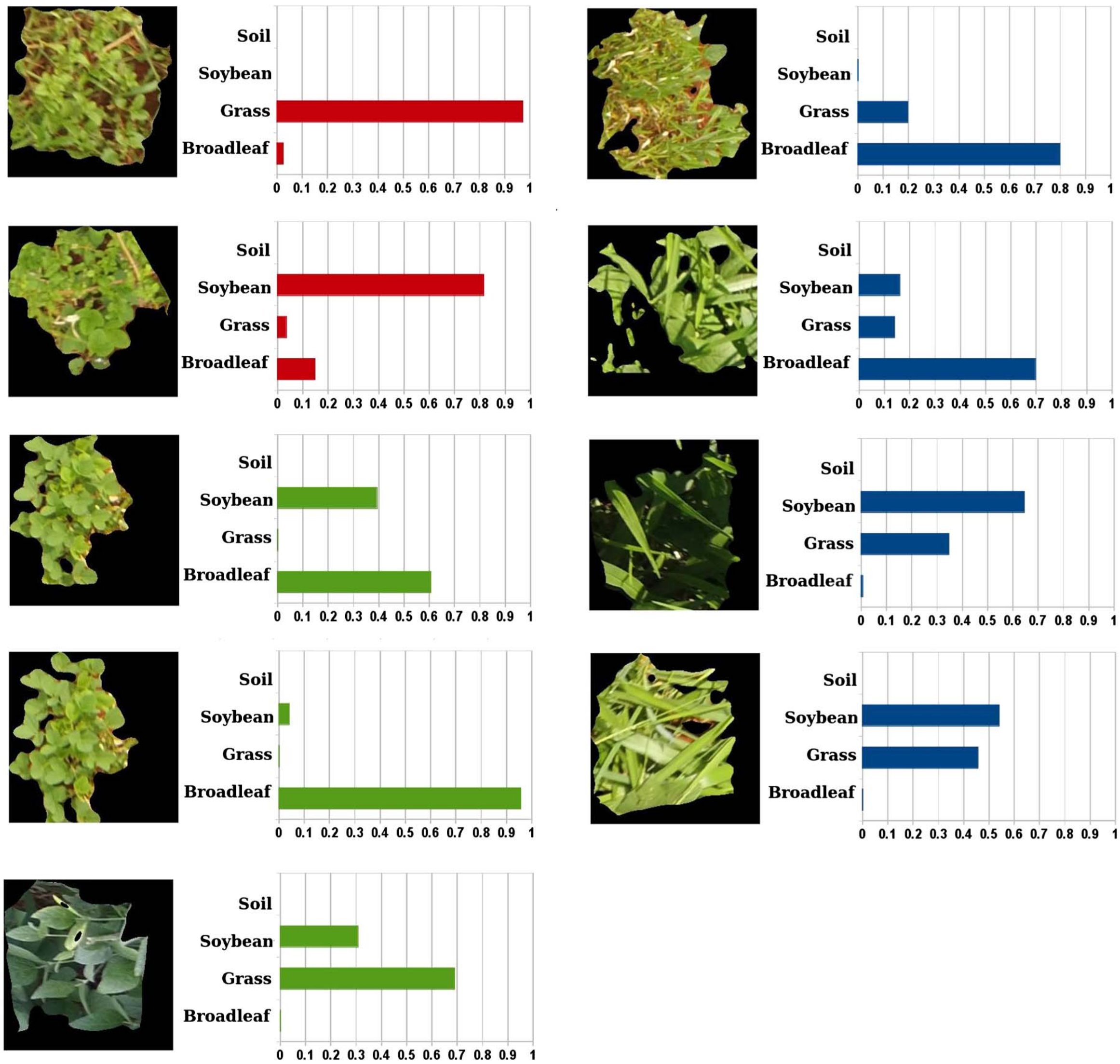

Fig. 7. Images classified incorrectly by the ConvNets in evaluation with balanced data. The red, green and blue colors on the figure represent, respectively, images of broadleaf weeds, soybean and grasses. The figure shows the probability in the classification of each image in the four classes of the problem. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Classification of ConvNet using the probability prediction as threshold. Using this approach a classification is considered valid only if its probability is higher than a predefined threshold.

	Classified		
Probability	Correctly	Incorrectly	Not considered
= 1	480	0	520
> = 0.999	860	0	140
> = 0.99	947	0	53
> = 0.98	963	0	37
> = 0.96	973	1	26
> = 0.94	978	2	20
> = 0.90	981	2	17
> = 0.75	985	4	11
> = 0.50	991	9	0

chosen by a specialist or through statistical analysis of the results.

3.2. Evaluation with unbalanced data

In this second evaluation, the performance averages had an increase for all algorithms analyzed, according to Table 4. In the case of algorithms compared to ConvNets, the increase was even more significant. It is an interesting behavior because the neural network was expected to require a large mass of data to achieve satisfactory results. However, it has already obtained results close to the ideal with a significantly smaller number of inputs, used in the previous evaluation, while the other classifiers needed a greater amount of data to approximate the result of the ConvNets.

In addition, despite the increase in performance averages, the accuracy and sensitivity to broadleaf weeds had a reduction in performance in some algorithms, even though it was trained a sampling of broadleaf weed images of similar size in the two evaluations. This behavior demonstrates how the unbalanced data can lead to an impact on the classification result. Classes with more trained elements may become more sensitive to identification in relation to classes with fewer trained elements, as appears to have occurred with broadleaf class. Even the neural networks have suffered from this problem, although in a less conclusive way.

3.3. Evaluation by extractors

A determinant factor in the efficiency of the classification of the

Table 4
Confusion matrix of all classifiers evaluated, for the group of 15,000 images of the image dataset. The bold values represent the weighted average precision and sensitivity of each classifier.

	Soil	Soybean	Grass	Broadleaf	Precision	Sensitivity
ConvNets						
Soil	650	0	0	0	1.000	1.000
Soybean	0	1406	2	0	0.995	0.998
Grass	0	4	696	4	0.997	0.988
Broadleaf	0	3	0	235	0.983	0.987
Weighted Avg.					**0.995**	**0.995**
Support Vector Machines						
Soil	649	0	1	0	0.997	0.998
Soybean	0	1392	13	3	0.980	0.989
Grass	2	25	669	8	0.974	0.950
Broadleaf	0	3	4	231	0.955	0.971
Weighted Avg.					**0.980**	**0.980**
Adaboost – C4.5						
Soil	650	0	0	0	0.995	1.000
Soybean	0	1395	9	4	0.982	0.991
Grass	3	14	686	1	0.972	0.974
Broadleaf	0	12	11	215	0.977	0.903
Weighted Avg.					**0.982**	**0.982**
Random Forest						
Soil	649	1	0	0	0.991	0.998
Soybean	0	1367	37	4	0.966	0.971
Grass	5	29	658	12	0.928	0.935
Broadleaf	1	18	14	205	0.928	0.861
Weighted Avg.					**0.960**	**0.960**

algorithms compared to the ConvNets is the quality of the attributes provided by the chosen extractors. To evaluate this impact, each classifier was analyzed using the attributes extracted by each algorithm individually. With this evaluation it is aimed to analyze the power of each extractor and if their use together could not have somehow hampered the performance of the classifiers. This evaluation was done using the unbalanced dataset.

As seen in Table 5 there was a noticeable drop in the performance of the classifiers when extractors were used individually. The Adaboost and Random Forest algorithms achieved their best results using the color features while the Support Vector Machine had its best result using the LBP extractor. Another interesting analysis shows that all the analyzed classifiers did not have good results when the GLCM and HOG extractors were used, with the HOG extractor having a much lower performance than others extractors. This behavior suggests that color and texture features are more suitable for this problem. While color is determinant in soil and plant discrimination, texture is an essential factor in the discrimination between soybean and weeds.

Given the high precision achieved by all classifiers compared, the need for the application of ConvNets, a structure that requires substantially more time and memory to train the dataset (Table 6), can be questioned. However, this last evaluation demonstrates how the strength of the collection of feature extractors used was determinant for these classifiers to present themselves in a competitive way against the

Table 5
Confusion matrix of classifiers compared to Neural Networks, with performance evaluated individually by extractor. The bold values represent the precision using all extractors and the best extractor evaluated for each classifier.

	All	GLCM	HOG	LBP	Colors
Support Vector Machines					
Precision	**0.980**	0.663	0.562	**0.899**	0.885
Adaboost – C4.5					
Precision	**0.982**	0.702	0.552	0.877	**0.941**
Random Forest					
Precision	**0.960**	0.697	0.550	0.865	**0.930**

Table 6
Training time and memory usage of the algorithms using CPU Intel (R) Core i7-5820 K CPU @ 3.30 GHz with GPU NVIDIA GeForce GTX TITAN X 12 GB, 32 GB of RAM and SSD. ConvNets training time and memory usage are in bold.

	Balanced Dataset		Unbalanced Dataset	
	Time	Memory	Time	Memory
ConvNets	**528.16 s**	**1715 MB**	**1785.71 s**	**1715 MB**
Support Vector Machines	0.78 s	329 MB	4.31 s	481 MB
Adaboost – C4.5	16.28 s	283 MB	110.14 s	1207 MB
Random Forests	2.33 s	623 MB	9.07 s	1217 MB

Neural Networks, a performance that was not achieved using the extractors individually.

Although it is possible to assume that using other combinations of feature extractors could achieve results even higher than the ConvNets, we must keep in mind that exhaustive tests or surveys with domain experts would be necessary until finding the best combination of extractors and classifiers for this problem in specific. Using ConvNets we have the advantage of giving up this part of the work, arriving at the solution in a more direct and also flexible way, after all a combination of extractors that works well for a certain problem or even a image dataset, can not necessarily be used in another problem.

Through the use of ConvNets we have a good chance to apply the same technique used in the identification of weeds in soybean crops to other types of crops only by modifying the composition of the training image dataset. This aspect can be considered a compensation factor in relation to the high costs of time and memory spent in the training of ConvNets, mainly because these costs are already being significantly reduced by the recent advances in hardware.

3.4. Image classification

In addition to the classification evaluations of the segments of our image dataset, through the Pynovisão software, it was possible to classify the images captured by the UAV in the soybean plantation. For this task, the software performs image segmentation using the SLIC algorithm and after this step it classifies each segment independently in one of the four defined classes.

To perform the classification we use the trained model for the evaluation with unbalanced data. To ensure that the segments generated in the test image were not identical to some segment used in the training of our algorithms, the SLIC segmenter was used as the compactness value 30, while in our image dataset generation the value 40. This also proves the ability of our model to adapt classification to variations in relation to the parameters it has been trained. In any case, it is noticeable that even with this variation part of the training information continues to be reused.

We performed the test with three images, corresponding to the months of December, February and March. To compare the results of the ConvNets, we performed the same experiment with the SVM. As these images were not manually annotated by a set of experts prior to automatic classification, so that a quantitative analysis of the result could be performed as a ground truth, classification analysis can be done in a qualitative analysis based on the visual result provided by the software.

It is important to emphasize that this analysis does not represent the evaluation of the ConvNets accuracy in weed detection, but rather the qualitative evaluation of the Pynovisão software developed in this work and its possible effectiveness in a real field application.

The first image, Fig. 8, contains segments of all classes evaluated. The weeds were correctly identified. However, there was a certain discrepancy between the algorithms analyzed in their descriptions. One of the possible causes of this behavior is related to the fact that several stretches of the crop contain the presence of both types of weeds,

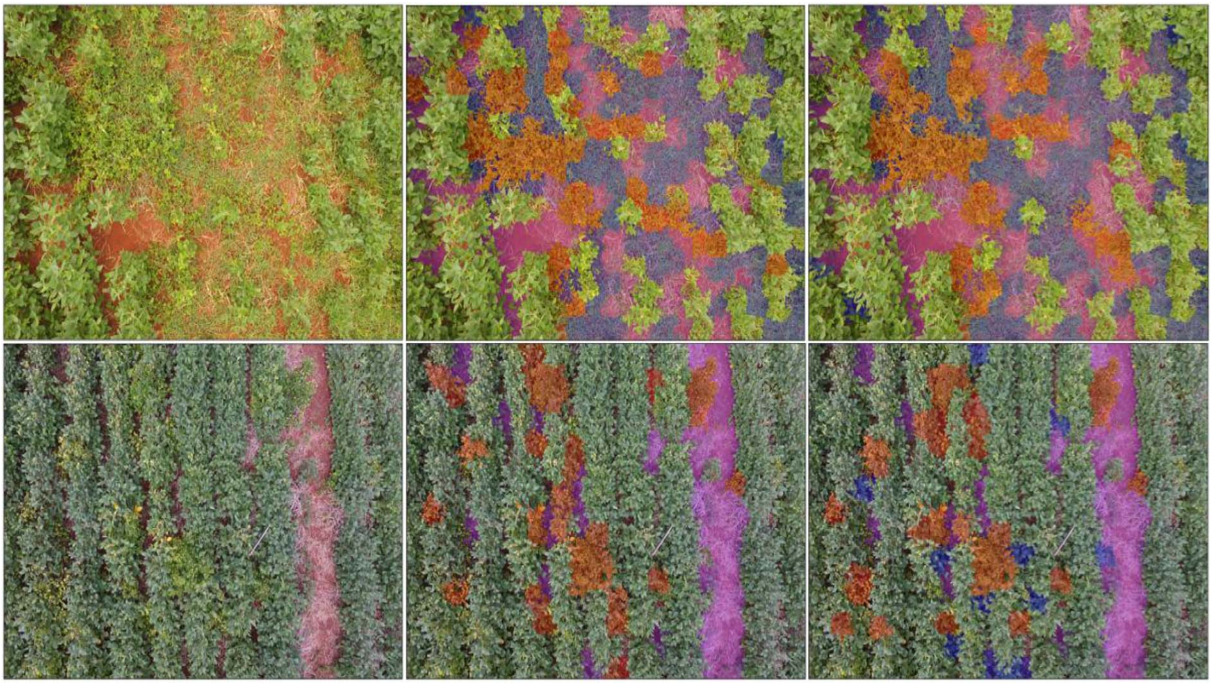

Fig. 8. From left to right, the original image, the image classified by the ConvNets and classified by the Support Vector Machine. Red represents broadleaf weeds, blue represents grasses and purple the soil. The soybean was kept in its original color. Each row represents a distinct day. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

causing several of the superpixels to encompass more than one class of the problem. Without the definition of a minimum threshold for the classification of a segment is considered valid, the software is obliged to provide classification to all segments, even without a high level of confidence. In this image both algorithms reached an accurate identification of the soybean segments.

In the second analyzed image, Fig. 8, corresponding to the plantation in the month of March, there were only examples of broadleaf weeds. SVM obtained 3.19% false positives from grass weeds, as can be seen in the blue segments in the lower right-hand corner of the image 8. The ConvNets, correctly, did not classify any segment of the image as grass. Both algorithms were able to detect broadleaf weed outbreaks in the image, despite classifying some segments as soybean.

The last analyzed image, Fig. 9, corresponding to the month of December, has some characteristics to be analyzed. It was photographed at a height above the standard used and with 16 × 9 resolution, unlike the 4 × 3 standard adopted in the images used in the generation of the database. Moreover, due to the problems in the acquisition of images of the month of January, the great majority of the images used in the creation of the dataset was of images corresponding to the reproductive stages of the soybean. This image represents soy in its vegetative state.

Even with these variations, the analyzed algorithms obtained a high accuracy of detection and discrimination of the weeds in the photo. Analyzing the quantitative results ConvNets detected 24.57% of grass and 19.24% of broadleaf weeds in the plantation area. The Support Vector Machine obtained a similar result, detecting 25.01% of grass and 22.36% of broadleaf weeds. However, by analyzing the Fig. 9, it is possible to identify that the difference of 3.12% between the two algorithms in relation to the broadleaf weeds correspond to false positives

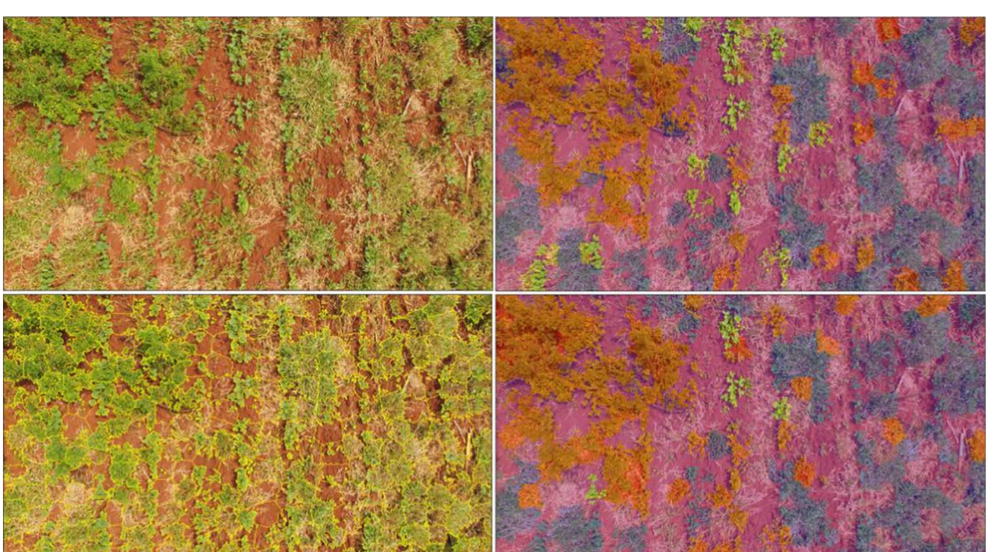

Fig. 9. Image segmented and classified by the software Pynovisão. In the upper row we have the original image and the image classified by ConvNets. In the bottom row the image segmented by the SLIC algorithm and classified by Support Vector Machines.

of the Support Vector Machine in relation to segments representing soybean.

Anyway it is possible to notice the low rate of false positives, in both algorithms, in the detection of weeds. The ConvNets classified with good sensitivity the segments of soybean. This demonstrates the software's ability to recognize soybean independent of its phenological stage. Although the soybean samples were mostly used at the reproductive stage, the software achieved a high precision and sensitivity rate in the classification of segments containing soybeans at the vegetative stage, which is the stage where weeds need to be identified and controlled to prevent losses.

4. Conclusion

In this work, it was developed a software that performs the detection of weeds in soybean crop images, as well as discriminating between grass and broadleaf weeds. The SLIC Superpixel algorithm was an efficient segmentation tool for images of plantations captured by UAVs, in addition to optimizing the time spent in the construction of the image dataset. The dataset built in this work is composed of more than fifteen thousand images of soil, soybean and weeds and will be made available. The use of ConvNets achieved excellent results, with accuracy higher than 98% in the classification of all classes. In the set of 15 thousand images, 99.5% average accuracy was obtained between all analyzed images. The compared algorithms also obtained good classification results, but the Neural Networks have the advantage that their results are not dependent on the choice of good feature extractors. In addition, the use of ConvNets can count on the recent benefits of the rapid increase in processing power and memory, which enable the training of large sets of images in a viable time.

As future works it would be interesting to address the evaluation with a image dataset covering a greater range of variables, such as different locations and height of image acquisition. Without the dependency on feature extractors, it is likely that the methodology used in this work can be extended to similar problems, involving another types of crops, with few adaptations. Given the high precision achieved in this work in the classification of weeds into grass and broadleaf weeds, it would also be interesting to evaluate the accuracy reached in the classification of weeds by species. While it is likely that a 99.5% precision rate may not be achieved in a real field application, since all the research was performed in a controlled environment, a close accuracy rate could be achieved in practice using a more diversified image dataset that represents the most varied types of soil and weeds.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. 34 (11), 2274–2282.

Ahmed, F., Al-Mamun, H.A., Bari, A.S.M.H., Hossain, E., Kwan, P., 2012. Classification of crops and weeds from digital images: A support vector machine approach. Crop Protect. 40, 98–104.

Ahonen, T., Hadid, A., Pietikainen, M., 2006. Face description with local binary patterns: Application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 28 (12), 2037–2041.

Arel, I., Rose, D.C., Karnowski, T.P., 2010. Deep machine learning – a new frontier in artificial intelligence research. IEEE Comput. Intell. Mag. 5 (4), 13–18.

Bengio, Y., 2009. Learning deep architectures for ai. Found. Trends Mach. Learn. 2 (1), 1–127.

Bradski, G., 2000. The opencv library. Dr. Dobb's J.: Software Tools Prof. Program. 25 (11), 120–123.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Cheng, B., Matson, E.T., 2015. A feature-based machine learning agent for automatic rice and weed discrimination. International Conference on Artificial Intelligence and Soft Computing, pp 517–527.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886–893.

Dos Santos, C.N., Gatti, M., 2014. Deep convolutional neural networks for sentiment analysis of short texts. COLING 69–78.

Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm. Icml 96, 148–156.

Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 36 (4), 193–202.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software: an update. ACM SIGKDD Explor. Newslett. 11 (1), 10–18.

Herrera, P.J., Dorado, J., Ribeiro, A., 2014. A novel approach for weed type classification based on shape descriptors and a fuzzy decision-making method. Sensors 14 (8), 15304–15324.

Hung, C., Xu, Z., Sukkarieh, S., 2014. Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a uav. Remote Sens. 6 (12), 12037–12054.

International: World soybean production, 2017. Soystats, <http://soystats.com/international-world-soybean-production/> (last visited July 26, 2017).

Ishak, A.J., Hussain, A., Mustafa, M.M., 2009. Weed image classification using gabor wavelet and gradient field distribution. Comput. Electron. Agric. 66 (1), 53–61.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678.

Kalchbrenner, N, Grefenstette, E., Blunsom, P., 2014. A convolutional neural network for modelling sentences. Available from: arXiv preprint <arXiv:1404.2188>.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, pp. 1097–1105.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems, pp. 396–404.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Liaw, A., Wiener, M., 2002. Classification and regression by randomforest. R News 2 (3), 18–22.

Platt, J., 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. Microsoft Reas.

Quinlan, J.R., 1993. C4. 5: Programs for Machine Learning. Morgan Kaufmann.

Rizzardi, M.A., Fleck, N.G., 2004. Métodos de quantificação da cobertura foliar da infestação das plantas daninhas e da cultura de soja. Ciência Rural 34 (1), 13–18.

Saha, D., Hanson, A., Shin, S.Y., 2016. Development of enhanced weed detection system with adaptive thresholding and support vector machine. Proceedings of the International Conference on Research in Adaptive and Convergent Systems, pp. 85–88.

Siddiqi, M.H., Lee, S., Kwan, A.M., 2014. Weed image classification using wavelet transform, stepwise linear discriminant analysis, and support vector machines for an automatic spray control system. J. Inform. Sci. Eng. 30 (4).

Silva, A.C., Lima, E.P.C., Batista, H.R., 2011. A importância da soja para o agronegócio brasileiro: uma anólise sob o enfoque da produção, emprego e exportação. Encontro de Economia Catarinense.

Soh, L.K., Tsatsoulis, C., 1999. Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. IEEE Trans. Geosci. Remote Sens. 37 (2), 780–795.

Tellaeche, A., Pajares, G., Burgos-Artizzu, X.P., Ribeiro, A., 2011. A computer vision approach for weeds identification through support vector machines. Appl. Soft Comput. 11 (1), 908–915.

Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., 2014. scikit-image: image processing in python. PeerJ 2, e453.

Voll, E., Gazziero, D.L.P., Brighenti, A.M., Adegas, F.S., Gaudencio, C.A., Voll, C.E., 2005. Dinâmica das plantas daninhas e práticas de manejo. Embrapa Soja.

Wallach, I., Dzamba, M., Heifets, A., 2015. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. Available from: arXiv preprint <arXiv:p1510.02855>.