Computers and Electronics in Agriculture 179 (2020) 105836

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Detection and classification of soybean pests using deep learning with UAV images

Everton Castelão Tetila [a,e,*], Bruno Brandoli Machado [b], Gilberto Astolfi [b,f], Nícolas Alessandro de Souza Belete [c,e], Willian Paraguassu Amorim [a], Antonia Railda Roel [e], Hemerson Pistori [b,e]

[a] Universidade Federal da Grande Dourados, Dourados, Mato Grosso do Sul, Brazil
[b] Universidade Federal de Mato Grosso do Sul, Campo Grande, Mato Grosso do Sul, Brazil
[c] Universidade Federal de Rondônia, Cacoal, Rondônia, Brazil
[e] Universidade Católica Dom Bosco, Campo Grande, Mato Grosso do Sul, Brazil
[f] Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul, Campo Grande, Mato Grosso do Sul, Brazil

ARTICLE INFO

Keywords:
UAV
Remote sensing
Soybean pests
Precision agriculture
Deep learning

ABSTRACT

This paper presents the results of the evaluation of five deep learning architectures for the classification of soybean pest images. The performance of Inception-v3, Resnet-50, VGG-16, VGG-19 and Xception was evaluated for different fine-tuning and transfer learning strategies over a dataset of 5,000 images captured in real field conditions. The experimental results showed that the deep learning architectures trained with a fine-tuning can lead to higher classification rates in comparison to other approaches, reaching accuracies of up to 93.82%. In addition, deep learning architectures outperformed traditional feature extraction methods, such as SIFT and SURF with Bag-of-Visual Words approach, the semi-supervised learning method OPFSEMImst, and supervised learning methods used to classify images, for example, SVM, k-NN and Random Forest. The results indicate that architectures evaluated can support specialists and farmers in the pest control management in soybean fields.

1. Introduction

Vegetable soybean (*Glycine max* [L.] Merrill) is an oilseed with good nutritional profile and important to the world economic participation. The nutritional quality of vegetable soybean is determined by its content of protein, unsaturated fatty acid, minerals, vitamins, isoflavone and other trace nutrients in the fresh seeds (Hou et al., 2011). From sowing to harvest, soybean cultivation is subject to the attack of defoliant pests such as insects and mollusks. Sampling methods, such as drop cloth, scanning net, visual plant examination, soil sampling and, more recently, pheromone-trapped have been employed to monitor the levels of pest control action in the soybean fields (Corrâa-Ferreira et al., 2012). Early detection of pests allows a more efficient application of pesticides, since inputs can be applied to the right quantity and locations, thus reducing production costs and the environmental impact resulting from the application of pesticides in the total area, in addition to contributing to human health and food safety (Tetila et al., 2019b).

As an alternative to manual sampling methods, technological innovations have helped control pests and increase food production in the field. UAVs equipped with high resolution cameras in data collection-missions are able to fly over a plant a few meters away and capture images rich in detail, which has helped to monitor the cultivation and harvesting of entire agricultural properties, with the aid of precision agriculture. Furthermore, the high cost of chemicals associated with low ecological impact actions lead to better precision agriculture practices. Thus, the use of UAVs in field crops has been considered an important tool to detect problems in the field, allowing experts and farmers to make better management decisions.

In recent years, several neural network architectures have become popular due to impressive results in image classification and problem detection. Keyvan and Jafar (2013) proposed an artificial neural network (ANN) with a 3 layers for identification of the insect Lepidoptera *Spodoptera exigua* from other species of pests. Similarly, an ANN was trained by Leow et al. (2015) for the classification of Copepod species - a

* Corresponding author.
E-mail addresses: evertontetila@ufgd.edu.br (E.C. Tetila), brunobrandoli@gmail.com (B.B. Machado), gilberto.astolfi@ifms.edu.br (G. Astolfi), nicolas.belete@unir.br (N.A.S. Belete), willianAmorim@ufgd.edu.br (W.P. Amorim), arroel@ucdb.br (A.R. Roel), pistori@ucdb.br (H. Pistori).

https://doi.org/10.1016/j.compag.2020.105836
Received 17 July 2018; Received in revised form 29 May 2020; Accepted 13 October 2020
Available online 1 November 2020

very important group of crustaceans in the marine chain. In (Yaakob and Jain, 2012) an ANN was combined with six different techniques of invariant moments to extract shape characteristics from the images used in the insect recognition task. Artificial neural networks were also designed in (Wang et al., 2012) for automatic identification of insect species at the level of their order. In the work of Wen et al. (2015) the authors trained a convolutional neural network to estimate the pose of the moths collected in the field. Combinations of texture, color, shape and local characteristics were extracted based on the specific pose of the butterfly and used as input to the deep learning model. In (Al-Saqer and Hassan, 2011) an ANN was used to recognize the presence of the *Weevil Red Palm* insect and distinguish it from other insects found in the palm habitat. Recently, Guoguo et al. (2017) proposed a pest recognition system based on image salience analysis and a deep learning model for the task of classifying insect species in the Chinese tea fields.

In all the works cited, the image acquisition does not cover the real field conditions, which provide various lighting conditions, such as sun reflection, cloud coverage, shadow and background variations. Furthermore, most of the authors did not compare the results with other approaches considered state-of-the-art, such as ResNet-50, VGG-19 and Inception-v3. In this context, an approach based on deep learning using images captured in real field conditions, under different conditions of lighting, object size and background variations was proposed to detect diseases and pests in tomato plants (Fuentes et al., 2017), in soybean (Amorim, 2019) and for the automatic counting of soybean pests (Tetila, 2019b). Machado et al. (2016) developed a mobile application called *BioLeaf* to measure the damage of soybean leaves caused by the insect herbivory based on two techniques, *Otsu segmentation* and *Bezier curves*. The application was tested in real-world images collected from soybean fields, and the tool can be used for different wide and narrow leaf crops. A review of the literature on classification of insects based on images, in which the questions that remain unresolved have been investigated, was presented in (Martineau et al., 2017).

Remote sensing methods using different types of optical technologies, such as RGB images (Tetila et al., 2017;Tetila et al., 2019a), acoustic sensors (Liu et al., 2017a), X-ray software (Chelladurai et al., 2014), thermography (Calderón et al., 2015; Oerke et al., 2006;Mahlein et al., 2012), ultraviolet (Liu et al., 2017b; Peruca et al., 2018), chlorophyll fluorescence (Calderón et al., 2015; Mahlein et al., 2012), LiDAR (Weiss et al., 2010) and multi-hyperspectral (Calderón et al., 2015; Yanan et al., 2014;Mahlein et al., 2012; Lu et al., 2018) have been proposed to capture field images in specific spectral bands to increase crop yield. In (Sirisomboon et al., 2009) the reflectance spectroscopy was investigated by means of a spectrometer ranging from visible light (VIS) to the near infrared (NIR) region (600–1100 nm) for the detection of defects (external and internal) in green soybeans caused by insects and diseases. Gedeon et al. (2017) described the mechanism of configuration and operation of an infrared optoelectronic sensor to detect soil microartropodes in the 0.4–10 mm size range and to estimate the body length (size) of soil microparthroids in field conditions. In (Chelladurai et al., 2014) hyperspectral imaging techniques in the near infrared (NIR) region and X-ray software were used to detect *Callosobruchus maculatus* infestation in soybeans - a storage pest that causes large storage losses in legumes.

Machine learning methods have also been proposed to detect insect species, such as bees (da Silva et al., 2015), common invertebrates (e.g., butterflies, grasshoppers) and molluscs (e.g., snails, slugs) (Liu et al., 2017a), besides counting white flies (Barbedo, 2014) and aphids (Maharlooei et al., 2017; Shajahan et al., 2016) in soybean leaves. UAV-based remote images were proposed to identify diseases in soybean (Tetila et al., 2017;Tetila et al., 2019a; Brodbeck et al., 2017) and also in citrus (Garcia-Ruiz et al., 2013). In (dos Santos et al., 2017) the authors used UAV-based remote images and a convolutional neural network to detect weeds in soybean, differentiating them between narrow leaves and broad leaves, in order to guide the application of herbicides. Pantazi et al. (2017) reported the detection and mapping of weeds using a hierarchical self-organising map and a multispectral camera mounted on a fixed wing UAV. However, no studies have been found in the literature that address the use of UAV images for the detection of soybean pests in the field.

In this paper, five deep learning architectures were compared for the task of detecting and classifying images of soybean pests collected in real field conditions. Initially, an image segmentation step using the SLIC superpixels algorithm (Achanta et al., 2012) was used to identify individual pest in the leaves of the plants obtained in the image acquisition step. During the inspection phase, aerial images were captured using a low-cost UAV well known in the market, DJI Phantom 4 Advanced model. Then an entomologist biologist labeled each pest image to identify its specific class and describe examples of each class. Our methodology evaluates five deep learning architectures and compares them with other machine learning methods. The proposed approach uses a set of 5,000 images, divided into 13 classes: (1) *Acrididae*, (2) *Anticarsia gemmatalis*, (3) *Coccinellidae*, (4) *Diabrotica speciosa*, (5) *Edessa meditabunda*, (6) *Euschistus heros* adult, (7) *Euschistus heros* nymph, (8) *Gastropoda*, (9) *Lagria villosa*, (10) *Nezara viridula* adult, (11) *Nezara viridula* nymph, (12) *Spodoptera* spp. and (13) without the presence of pests - to measure acuracy, training time and learning error of the deep learning architectures in the task of classifying soybean pests. These pests species are common in several regions of soybean crops around the globe, and whether it is not managed adequate and on time, they often cause a huge loss in the grain productionin of cultivars, such as soybeans, corn, wheat and beans.

2. Simple Linear Iterative Clustering (SLIC)

The Simple Linear Iterative Clustering algorithm (SLIC) groups regions of pixels in the 5-D space defined by L, a, b (values of the CIELAB color scale) and the coordinates x and y of the pixels. An input image is segmented into atomic regions, defining the number k of superpixels with approximately $\frac{N}{k}$ pixels, where N is the number of pixels in the image. Each region composes an initial superpixel of dimensions $S \times S$, where $S = \sqrt{\frac{N}{k}}$. The centers of the superpixel clusters $C_k = [l_k, a_k, b_k, x_k, y_k]$ with $k = [1, k]$ are chosen, spaced on a regular grid to form clusters of approximate size S^2. The centers are moved to the lowest gradient value over a neighborhood of 3×3 pixels, avoiding centroid allocation in edge regions that have noisy pixels. Instead of using a simple Euclidean norm in space 5D, a distance measure D_s is defined as follows:

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \quad (1)$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad (2)$$

$$D_s = d_{lab} + \frac{m}{s} * d_{xy} \quad (3)$$

where D_s is the sum of the distance d_{lab} (Eq. (1)) and the distance d_{xy} (Eq. (2)), normalized by the interval S. The parameter m corresponds to the superpixel compactness control; the greater the value, the more compact the clustering is in terms of spatial proximity. Each pixel of the image is associated to the closest centroid and, after all pixels are associated to a centroid, a new center is calculated with the *Labxy* vector of all superpixels belonging to the group. At the end of the process, some pixels may be connected to a group incorrectly, so the algorithm reinforces connectivity in the last step by assigning the pixels alone to the largest neighboring groups (Achanta et al., 2012).

3. Deep learning

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have considerably improved the

state of the art in speech recognition (Hinton et al., 2012), visual object recognition (Wang and Yeung, 2013), object detection (Girshick et al., 2014), segmentation (Long et al., 2015), video classification (Karpathy et al., 2014) and many other domains. Deep learning models are capable of learning large data sets by using the an iterative algorithm, back-propagation, that indicates how the model should change its internal parameters in order to learn feature representations in each layer (LeCun et al., 2015).

Several deep learning architectures have been proposed in the last decade. These architectures are usually evaluated and compared using well-know datasets such as ImageNet (ImageNet, 2016). Improvements in network architectures often transfer significant performance gains to a wide variety of application domains that rely increasingly on learned visual representations. Next, we present five deep learning architectures widely used in computer vision tasks.

VGGNet (Simonyan and Zisserman, 2014) uses an architecture with very small convolution filters (3*x*3), which show that a significant improvement over prior configurations can be reached by pushing the depth to 16–19 weight layers. These findings achieved the first and second place in the localization and classification tracks, respectively, in the ImageNet Challenge 2014 (Russakovsky et al., 2015). In addition, the VGG-16 and VGG-19 models generalize well to other datasets, obtaining good performance results on computer vision.

Microsoft's Residual Networks (ResNet) (He, 2016) presents a residual learning framework to ease the training of networks that are substantially deeper than those used previously. Layers are explicitly reformulated as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. These residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset, residual nets with a depth of up to 152 layers (8 times deeper than VGG nets) were evaluated, but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task (Russakovsky et al., 2015).

GoogleNet or Inception architecture (Szegedy et al., 2016) explores ways to scale-up networks aiming at efficient computation by suitably factorized convolutions and aggressive regularization. The computational cost of the Inception network is also much lower than VGGNet or its higher performing successors (He et al., 2015). The Inception architecture (Szegedy et al., 2015) is designed to work well even under strict memory and computational cost constraints. For example, GoogleNet employs around 7 million parameters, representing a nearly 9-fold reduction over its predecessor AlexNet, which uses 60 million parameters. In addition, VGGNet employs about 3 times more parameters than AlexNet. This made viable the use of Inception networks in big data scenarios where a large amount of data needs to be processed at a reasonable cost or scenarios where memory or computational capacity are inherently limited. However, the complexity of the Inception architecture makes it difficult to perform changes to the network. If the architecture is increased spontaneously, large parts of computational gains can be immediately lost. This makes it much harder to adapt it to new use cases while maintaining its efficiency.

Xception (Chollet, 2017) is a convolutional neural network architecture that uses 36 convolutional layers. These layers are structured in 14 modules, all with linear residual connections around them, except for the first and last module. First, the data passes through the input stream, then through the average stream that is repeated eight times, and finally through the output stream. Depth-separable convolution layers with residual connections make the architecture very easy to define and modify; takes only 30 to 40 lines of code using a high level library, such as Keras (Chollet, 2015) or TensorFlow-Slim (Guadarrama and Silberman, 2016), not very different from an architecture like VGG-16 (Simonyan and Zisserman, 2014), but rather of architectures such as Inception V2 or V3 that are much more complex to set up. Xception has the same number of parameters as Inception V3, but the former is better in the ImageNet dataset due to more efficient use of model parameters.

4. Proposed approach

This section presents a computer vision approach to identify images of soybean pests collected in real field conditions. The proposed approach adopts the SLIC Superpixels method to segment the pests in the images. The SLIC method employs the k-means (Hartigan and Wong, 2013) algorithm for the generation of similar regions, called superpixels. A schematic diagram of the proposed system is shown in Fig. 1. It illustrates the methodology that consists of five steps: (a) Image acquisition, (b) SLIC segmentation, (c) Image dataset (d) Features learning, and finally (e) Classification of pests. Initially, a 2 m high-altitude flight inspection was conducted with the UAV in the soy fields to capture planting images (see step (a) in Fig. 1). These images were segmented using the SLIC superpixels method (Fig. 1 (b)). The k parameter of the algorithm refers to the number of superpixels in the image and allows to control the superpixels size. The parameter $k = 200$ was adjusted and corresponds to the approximate segmentation size of the largest pest in the image. In this case, the insect pests of the *Spodoptera* spp. and *Anticarsia gemmatalis* species.

The m parameter corresponds to the compactness control of the generated regions. We tested different m-values to segment pests in the images because this directly impacts the shape of the superpixel. Fig. 2 shows some superpixel images with different values of m: 10, 50 and 100. According to the figure, square superpixels mean a high compression value for the parameter m and a very high value (e.g., $m = 100$) does not correctly identify the edges of objects, on the other hand, a low value (e.g., $m = 10$) deforms the superpixel more. Thus, we set the parameters $k = 200$ and $m = 50$ to segment the pests in the images, defined by adherence to the size and compactness limits of the SLIC algorithm.

After segmentation of the image with the SLIC method, each superpixel was visually annotated by a specialist to compose a superpixel image dataset for training and testing of the system, see step (c) of Fig. 1. In this case, an entomologist biologist was responsible for evaluating the representativeness of the sample for the statistical analysis. Subsequently, a convolutional neural network was trained to learn the visual patterns of the superpixel images (see step (d) of Fig. 1) and to classify the images of soybean pest species (see step (e) of Fig. 1). The post-processing stage shows the acuracy result, the learning error and the training time of each machine learning architecture evaluated by our computer vision system.

5. Materials and methods

An experimental area of 2 hectares was planted with conventional soybean cultivars and no application of pesticides. The agricultural area shown in Fig. 3 is located in the experimental farm of the UFGD, located in the municipality of Dourados-MS, Brazil, with geographic coordinates 22°13′57.52″ South latitude e 54°59'17.93" West longitude.

Two different approaches were used to collect images of pests present in the experimental field. First, a smartphone camera equipped with the 12-megapixel resolution IMX260 sensor was used. A total of 5,000 JPG images were collected in different days and climatic conditions, between 8 h and 10 h and 17 h and 18 h:30 min, during the reproductive phenological stages (R1 to R6) of soy during the harvest from September 2017 to February 2018. In the experimental field, we observed that insect exposition at the plants top usually occurs at the beginning of the day or at the end of the afternoon, reinforcing the recommendation that planting canopy insect sampling be carried out, preferably, during periods under milder temperatures of the day, as reported in (Corrâa-Ferreira et al., 2012). These images were captured by the researcher on site, using the camera 50*cm* apart on the target of interest and a 0° angle of the camera relative to the ground. The targets in this case correspond to the defoliation pests that cause economic damage when found at high concentration levels in the soybean fields. Then, each image was annotated in order with the support of an entomologist biologist, thus constructing a collection of references of superpixels to the set of

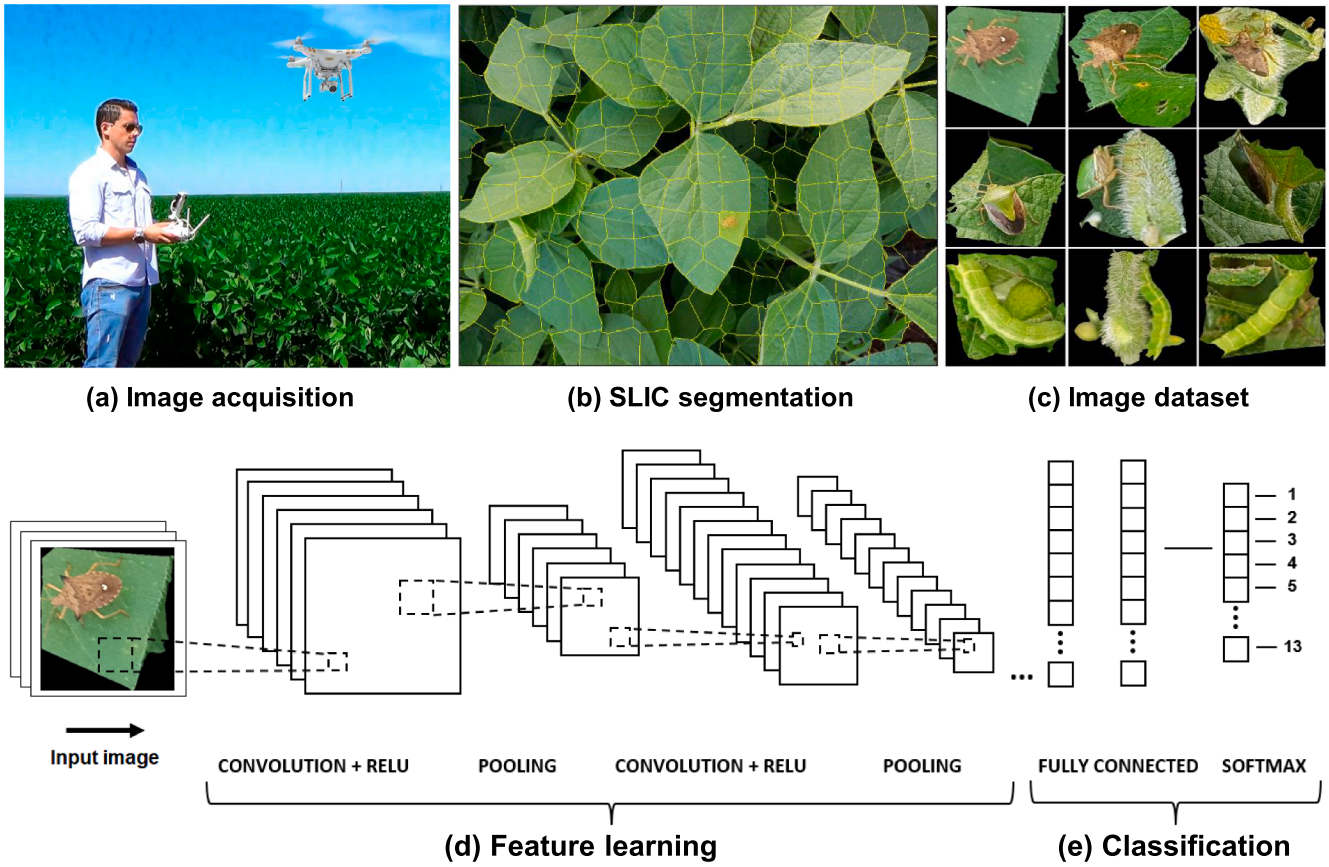

Fig. 1. Proposal of a computer vision system to identify soybean pests using deep learning with UAV images.

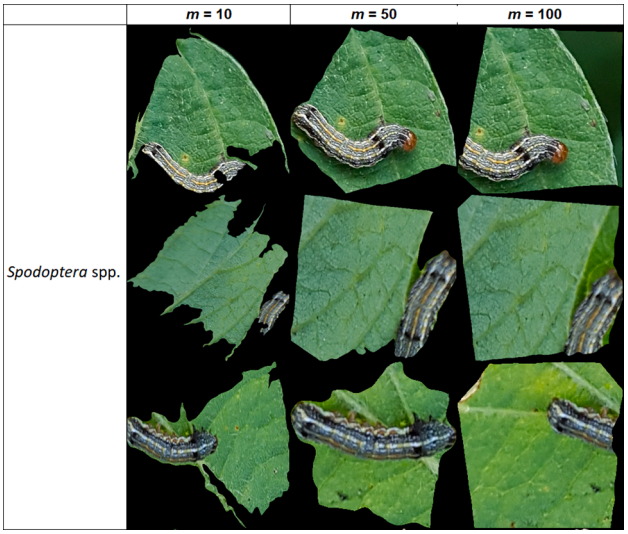

Fig. 2. Superpixel images with different values of *m*: 10, 50 and 100.

Fig. 3. Aerial view of the experimental area used for soybean planting.

training and test images of the system (see Fig. 4), called INSection 5K13C and available in (Tetila, 2018).

The uneven number of samples used in the training and test set reflects the occurrences of each pest species in the field. Balancing the training set is very common and usually gives good results, but we believe that the unbalanced training set does not affect the data representations, since the test set should remain unbalanced to avoid overfitting, reflecting the reality of the field where the occurrences of each pest species is not balanced.

In the second approach, 300 aerial images (JPG) were captured at 2 m of height of the plantation, using the UAV DJI Phantom 4 Advanced, equipped with a 1-inch Sony CMOS camera and 20 megapixels of resolution. In this case, the high-altitude flight of two meters was chosen because smaller values cause the displacement of the plants due to the wind generated by the rotors, which substantially modifies the initial positioning of the pests. On the other hand, for higher values the size of the pests in the images is gradually reduced and, consequently, the image resolution of the pests decreases.

Although we have not evaluated the drone's vibration in the quality of UAV images in this work, we believe that the influence is minimal

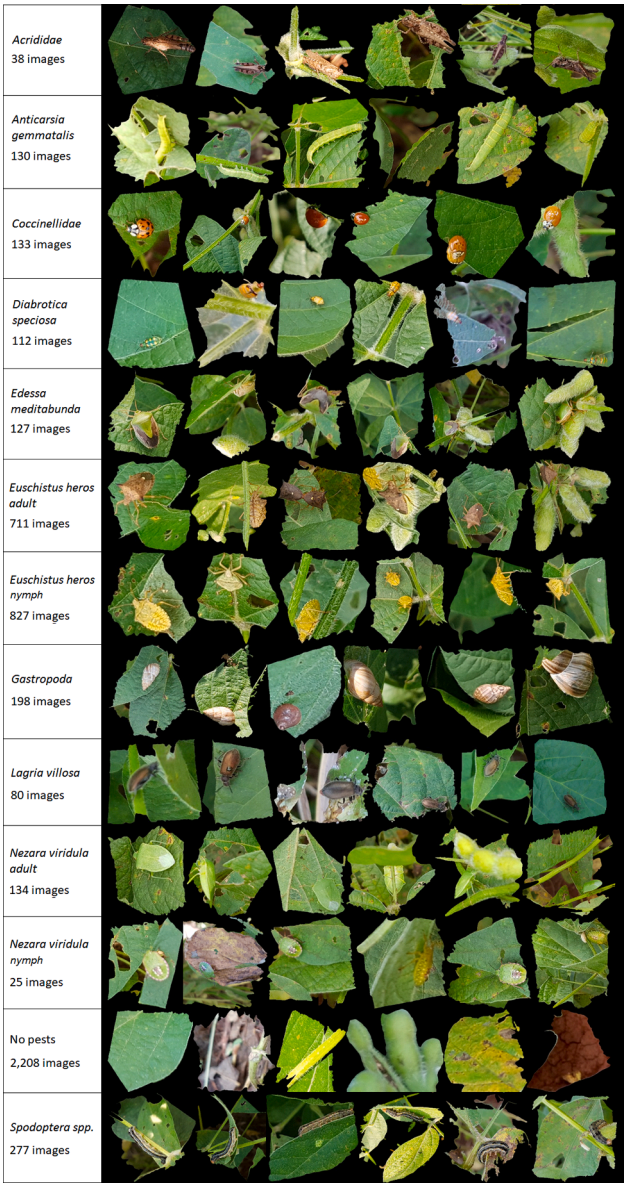

Fig. 4. Superpixel image samples from our image dataset, divided into pest species of the soybean crop and number of images per class. The images were collected for the real field conditions, which provide various lighting conditions, such as sun reflection, cloud coverage, shadow, size and positioning of objects, occlusion, background variations, mating and development phases.

because the UAV camera usually has a very fast shutter speed for capturing images (e.g., Phantom 4 - Electronic Shutter Speed: 8s to 1/8000s). We have published a paper in (Tetila et al., 2017) that addresses identification of soybean diseases using UAV images. The results indicate that there is a great influence of the image resolution in the identification system. The greater the distance between the camera and the targets of interest, the lower the accuracy of the identification system.

Each UAV image add geographic identification metadata (geotagging). This data usually consists of latitude and longitude coordinates, altitude, image data, camera data and time stamp. Geotagging can help users find a wide variety of location-specific information from a device, such as a smartphone or UAV. For instance, someone can find a pest infestation area in soybean by entering the latitude and longitude coordinates of the images taken near a given location into a suitable image search engine for better management decisions. UAV images do not comprise our training and test set because of the limitation found to visually identify pests in the images. Despite this, we discussed in Section 6.3 the detection of soybean pests with UAV images for higher altitudes.

The wide period of image acquisition during the reproductive phenological stages of soybean covers the real field conditions that provide various lighting conditions, such as sun reflection, cloud coverage, shadow and background variations. Other works that use deep convolutional neural networks for pest recognition have also been proposed, showing good performance on different crops. However, usually authors use images of pests previously collected in the field and captured by a camera in the laboratory, where the background lighting and reflection can be well controlled, but very different from a real scenario.

This is different from our soybean pest dataset, which has images collected directly on the site on different days (sunny, slightly cloudy, very cloudy). Furthermore, these images contain unwanted climatic variations that can negatively impact the learning of visual patterns by deep convolutional neural networks. For example, Fig. 5 shows the activation filters recognizing some undesirable visual patterns, including sun reflection, shadow, occlusion, herbivory and complex background, which do not correspond to the object of interest (pests), but that can negatively influence the performance of the ANN.

For image classification, deep learning architectures are trained with labeled images in order to learn how to classify them according to visual patterns. We use open source implementations of Xception, Inception-v3, VGG-16, VGG-19 and Resnet-50 architectures that are provided as part of the Keras module. We divided the data into 70% for training and 30% for testing. In the experiments, we used the following input parameters. The input image width and height were equally set in 256. The batch size was set 16 images for training and the number of epochs was used 50. We used the SGD optimizer with learning rate of 0.0001 and momentum of 0.9 (accelerate SGD in the relevant direction and dampens oscillations).

We applied data augmentation to increase the amount of data by applying rotation, rescaling, scrolling and zooming operations. This technique aims to reinforce the rotation invariance and scale invariance in the classification task, since the images are captured by the drone at different angles and scale. We also kept the same parameters for the data augmentation: rescale = 1./255 (multiplication factor for each pixel of the image); horizontal_flip = True (randomly alternates images horizontally); fill_mode="nearest" (points outside the input limits are filled according to the nearest direction); zoom_range = 0.3 (magnification factor); width_shift_range = 0.3 (horizontal shift factor); higth_shift_range = 0.3 (vertical displacement factor); rotation_range = 30 (image rotation factor).

We used two strategies of fixation rates of the neural networks with the weights obtained in the ImageNet dataset in order to evaluate each architecture and its behavior during the training process.

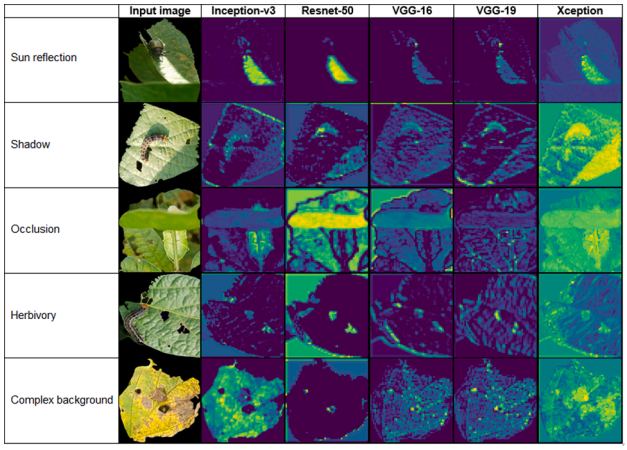

Fig. 5. Activation filters recognizing some undesirable visual patterns: sun reflection, shadow, occlusion, herbivory and complex background.

- **Transfer Learning:** given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S e T_S, where $D_S \neq D_T$, or $T_S \neq T_T$ (Pan and Yang, 2010). In this formulation, the knowledge acquired in a given task in a particular domain can be used to improve the learning of the predictive function in another task, in another domain.
- **Fine-tuning:** refers to re-using parameter values estimated on potentially large datasets as initialization in applications with limited access to labeled data (Kading et al., 2016). This strategy not only replaces and retrains the classifier in the dataset, but also fine-tunes the pre-trained neural network weights with the back-propagation algorithm. It is possible to tune all layers of the neural network or keep some of the previous layers fixed and only adjust the top-level part of the network.

In order to statistically evaluate the potential of the architectures for the classification of pests images in the soybean fields, we defined four different training strategies using the fine-tuning approach with the weights obtained from ImageNet ranging from 100% to 25%, with a 25% step, to the network layers. We also trained the complete network with the weights initialized randomly, in addition to the transfer learning approach with the weights obtained from ImageNet. In our experiments, we used five architectures: Inception-v3 (Szegedy et al., 2016), VGG-16 (Simonyan and Zisserman, 2014), VGG-19 (Simonyan and Zisserman, 2014), ResNet-50 (He, 2016) and Xception (Chollet, 2017).

In the classification task, we submit the deep learning architectures to the captured images. Three metrics were used to evaluate the performance of the architectures: acuracy, training time and learning error. In order to identify if the trained architectures differ statistically in relation to performance, we used the ANOVA hypothesis test in RStudio. A level of significance of 0.05 was used to rule out the null hypothesis. Then, a post-test was performed with the boxplot diagram to analyze the variation of the observed data and *p*-values were reported with the Tukey test.

In all our experiments we used a workstation with the hardware configurations described in Table 1.

6. Results and discussion

This section describes the results obtained by the proposed approach, followed by a discussion.

6.1. Classification evaluation

Fig. 6 shows the accuracy results obtained by the deep learning architectures in the test set, considering the values of Table 2. The highest absolute value of accuracy obtained by each architecture is highlighted in the table. Table 2 also shows the learning error and the total training time, in seconds, to construct the classification model. The time results of the Table 2 are referring to the hardware specifications given in Table 1. Executions in different machine configurations may interfere in the results presented.

In our experiments, the Resnet-50 architecture obtained the highest absolute value of accuracy (93.82%), followed by Inception-v3 (91.87%), VGG-16 (91.80%), VGG-19 (91.33%) and Xception (90.52%) architectures. The VGG-16 architecture obtained less training time, followed by the VGG-19, Resnet-50, Inception-v3 and Xception architectures. ANOVA test results indicate no evidence of statistically significant difference in average performance of the architectures tested at a significance level of 5% using the accuracy as metric (*p*-value = 0.0732).

We recently published a paper in (Tetila, 2019b) that addresses automatic pest counting in the field. In this work, we use a dataset with 10,000 images of soybean pests distributed in 7 classes of interest. Although the two datasets were created in different harvest times (2017/18 and 2018/19), they have the same problem classes (soybean pests). The results of the two works - one using 5 K dataset and the other using 10 K dataset - indicate that there is no significant difference (loss of accuracy) in the classification task among the tested deep learning architectures. For example, Resnet-50 using 100% fine-tuning reached an accuracy of 93.78% with the 10 K dataset, against 93.55% with the 5

Table 1
Workstation hardware technical specifications.

Hardware	Technical specification
Processor	Intel Core i7-6800 K 3.40 GHz 15 MB (6 N, 12T)
Graphics Cards	Geforce GTX1070 8 GB 1920 cuda cores
RAM memory	16 GB Kingston DDR4 2400 MHz
Storage	SSD 120 GB 2.5-SATA III Kingston UV400

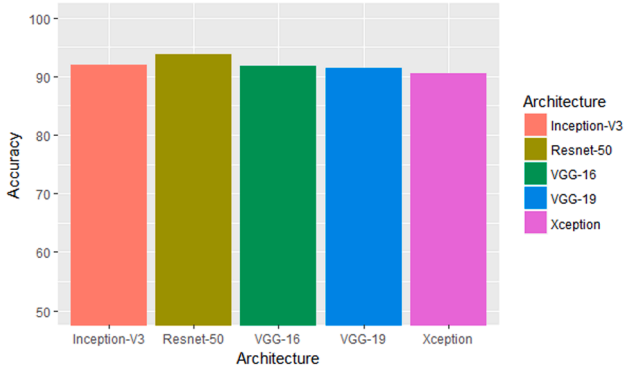

Fig. 6. Highest absolute value of accuracy obtained by each architecture.

Table 2
Performance metrics used to evaluate deep learning architectures.

Architecture	Training strategy	Training time (s)	Accuracy (%)	Learning error
Inception-v3	FTuning100%	5,066.99	91.26	0.3195
Inception-v3	FTuning75%	5,077.51	91.60	0.3297
Inception-v3	FTuning50%	5,064.74	91.80	0.3136
Inception-v3	FTuning25%	5,077.79	**91.87**	0.3064
Inception-v3	TrLearning	4,504.44	61.49	1.2493
Inception-v3	NoTrLearning	4,633.00	55.91	1.4025
Resnet-50	FTuning100%	4,977.64	93.55	0.2535
Resnet-50	FTuning75%	4,981.61	93.48	0.2564
Resnet-50	FTuning50%	4,968.79	**93.82**	0.2410
Resnet-50	FTuning25%	4,975.51	92.88	0.2684
Resnet-50	TrLearning	4,885.96	64.85	1.1051
Resnet-50	NoTrLearning	4,575.90	67.34	1.0690
VGG-16	FTuning100%	4,884.36	**91.80**	0.3098
VGG-16	FTuning75%	4,891.02	90.86	0.3577
VGG-16	FTuning50%	4,895.05	90.59	0.3633
VGG-16	FTuning25%	4,887.41	91.26	0.3722
VGG-16	TrLearning	4,858.54	51.81	1.5225
VGG-16	NoTrLearning	4,470.05	45.90	1.6980
VGG-19	FTuning100%	4,904.61	90.19	0.3625
VGG-19	FTuning75%	4,904.16	91.26	0.3422
VGG-19	FTuning50%	4,909.60	**91.33**	0.3241
VGG-19	FTuning25%	4,910.54	90.66	0.3562
VGG-19	TrLearning	4,883.55	50.47	1.6162
VGG-19	NoTrLearning	4,486.60	44.29	1.8172
Xception	FTuning100%	5,347.77	89.65	0.3776
Xception	FTuning75%	5,330.41	**90.52**	0.3283
Xception	FTuning50%	5,364.83	89.92	0.3815
Xception	FTuning25%	5,357.58	90.46	0.3740
Xception	TrLearning	4,513.90	65.52	1.1209
Xception	NoTrLearning	5,193.43	74.60	0.8430

K dataset. However, in our experiments we did not evaluate the loss of accuracy using a dataset with few images.

Certainly, there is an accuracy evolution curve according to the number of different images captured. We believe that the curve of the evolution is high in the initial phase and, as we increase the number of images proportionately, it decreases until it reaches stability. The ideal number of images for a dataset is specific to the problem; classification problems considered simpler can generally have a smaller set of training and test than the most complex, for example, images collected under various lighting conditions, such as sun reflection, cloud coverage, shadow, size and positioning of objects, occlusion and background variations.

Regarding the unwanted conditions that hinder the classification task, we believe that the training and testing set must be representative enough to reflect these variations; otherwise, the system will not have learned the features sufficiently to generalize well other datasets. In addition, data augmentation has been an excellent technique for increasing the dataset, applying rotation, resizing, scrolling and zooming operations. This technique aims to reinforce the rotation invariance and the scale invariance in the classification task. Even so, the dataset needs to be sufficiently representative according to the problem; otherwise, the system will not have learned the features for the classification task.

Fig. 7 shows the acuracy results of each deep learning architecture with the median value highlighted in the boxplot diagram. The diagram also shows the range of performance variation obtained by each architecture. The Resnet-50 architecture presented the highest absolute value for the median; Resnet-50 and Xception presented data dispersion in the best value range for acuracy in comparison to the other architectures.

Similarly, Fig. 8 shows the acuracy results and the interval of the performance variation of each training strategy. According to Fig. 8, the fine-tuning strategies (FTuning25, FTuning50, FTuning75 and FTuning100) presented greater absolute value for the median and dispersion of data in the best value range for acuracy in comparison to trained strategies with or without transfer learning (TrLearning and NoTrLearning). All comparisons between fine-tuning and transfer learning strategies resulted in p-values $<= 0.0000002$. Therefore, it is possible to reject the null hypothesis with the level of significance of 0.05 and to conclude that there is a statistically significant difference of acuracy between the strategies trained with fine-tuning and transfer learning. On the other hand, there is no significant difference between fine-tuning strategies (p-values $>= 0.9999997$) or between transfer learning and no transfer learning (p-value $= 0.9992375$).

We emphasize the importance of correctly choosing the strategy used in training deep learning architectures. For example, in Table 2 the VGG-16 architecture trained NoTrLearning strategy obtained acuracy of 45.90% against 91.80% using FTuning100% strategy, resulting in a difference of 45.90%. The training NoTrLearning considers complete network training with weights initialized at random. The NoTrLearning

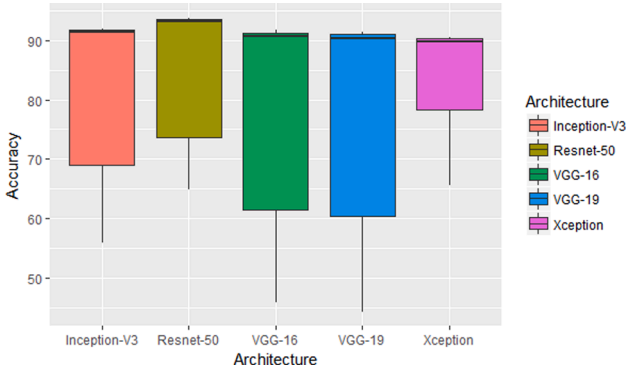

Fig. 7. Boxplot diagram comparing the accuracy results of each deep learning architecture.

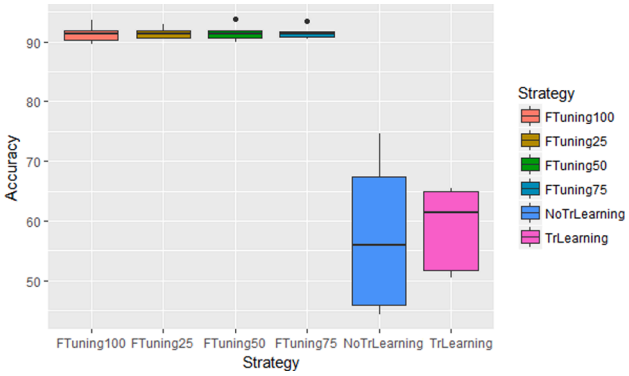

Fig. 8. Boxplot diagram comparing the accuracy results of each deep learning architecture.

results of Table 2 indicate that the accuracy of this strategy is lower because it does not re-use the values of the pre-trained parameters in potentially large dataset (e.g., ImageNet) as initialization in new applications. In the case of fine-tuning, this strategy not only replace and retrain the classifier in the dataset but also fine-tune the pre-trained neural network weights with the backpropagation algorithm. It is possible to tune all layers of the neural network (FTuning100%) or keep some of the previous layers fixed (e.g., FTuning50% or FTuning25%) and only adjust the top-level part of the network. This is motivated by the observation that earlier layers of a neural network learn more generic features (e.g., edge detectors or color detectors) that may be useful in many tasks, but the latter layers of the network become progressively more specific to the details of the classes contained in the original dataset.

In addition, the results of TrLearning in Table 2 show that the deep learning architectures trained with the weights obtained from ImageNet did not efficiently detect the visual patterns corresponding to soybean pests due to the complex background and lighting conditions. In contrast, fine-tuning achieved the highest classification rates compared to other training strategies. Together, our results show that the deep learning architectures with fine-tuning computed weights generalize well in soybean pests datasets.

We can see in Fig. 9 the result of applying the filters in the intermediate convolutional layers of the deep learning architectures. Note that even when the pest covers a small fraction of superpixels, convolutional filters can capture useful features that identify pests in the superpixels. The pixels corresponding to the pests are able to activate neurons, even when they occupy a small part of the superpixel, offering a good idea of what is happening internally in the convolutional layers.

Fig. 10 presents the confusion matrix of the deep learning architecture Resnet-50, trained with Fine-tuning 50%, since this architecture provided the highest absolute value of acuracy among the trained architectures. According to the figure, the *Euschistus heros* adult and *Euschistus heros* nymphs, represented by the letters F and G, obtained a higher number of instances classified incorrectly due to the difficulty of discriminating the nymph phases (from 1° to 5° instar) of the adult phase of the insect development cycle. These two classes comprise the same species of the insect *Euschistus heros*, which contributes to the great similarity of the visual patterns existing between these classes.

6.2. Comparison with other machine learning methods

In this experiment, the proposed approach is compared with other machine learning methods: local descriptors SIFT (Lowe, 1999) and SURF (Bay et al., 2008), supervised learning methods (SVM, *Random Forest*, J48, *Naive Bayes*, k-NN and *Adaboost*) and semi-supervised OPFSEMImst (Amorim et al., 2016). For this purpose, we used the same implementation and used the same dataset of soybean pest

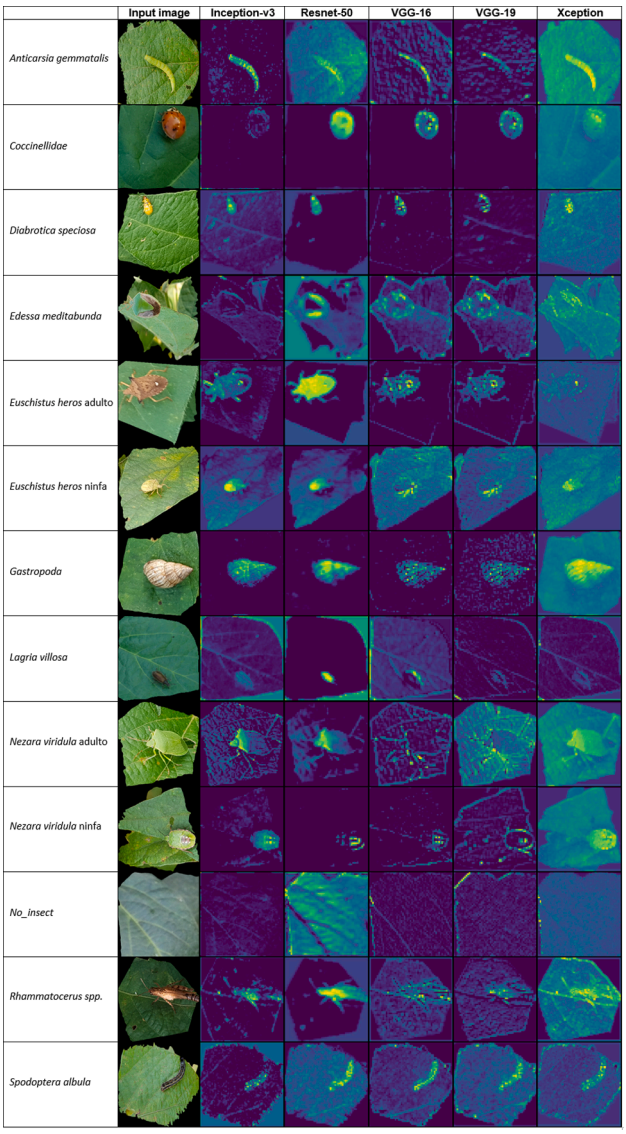

Fig. 9. Result of applying the filters in the intermediate convolutional layers of the deep learning architectures.

presented in Section 5. We also varied the parameter k with values 25, 50 and 100 for each local descriptor in order to define the number of visual words used in the dictionary of the *Bag-of-visual Words* approach. Table 3 shows acuracy and training time for each deep learning architecture and compares them with other machine learning methods.

As seen in Table 3, the deep learning architectures overcame all local descriptor methods (for all k values) and all supervised learning methods. The SIFT and SURF local descriptor methods provided 52.13% and 50.73% for 100 visual words, respectively. For supervised learning methods, SVM and Random Forest reached 60.46% and 56.42%.

The semi-supervised system OPFSEMImst of Amorim et al. (2016) was also trained using the same implementation of the authors and the same dataset presented in Section 5, divided into 70% for training (Z_1) and 30% for the test (Z_2). The set Z_1 was divided into Z_1' (labeled or supervised set) and Z_1'' (unlabeled or unsupervised set) for application of the semi-supervised process. OPFSEMImst uses the supervised training set and propagates the most strongly connected labels to the unsupervised set. After the propagation, a supervised classifier is generated and evaluated on the test set Z_2. Among propagation strategies, the highest absolute value of acuracy provided 53.61% training 70% of the supervised set ($Z_1' = 70\%$) and 30% of the unsupervised set ($Z_1'' = 30\%$).

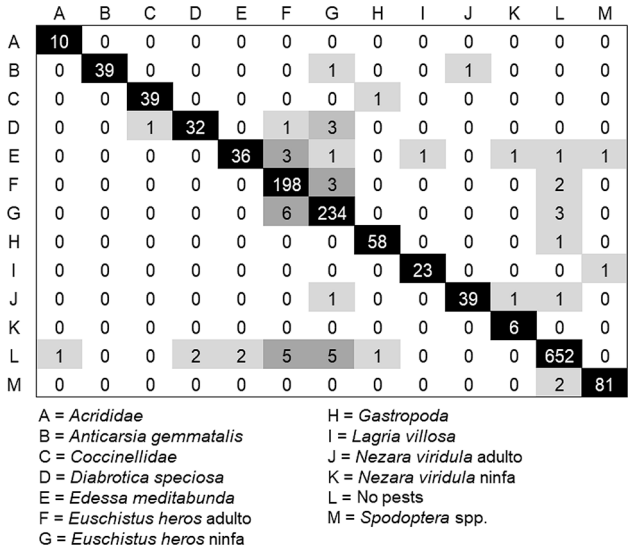

Fig. 10. Confusion matrix of the Resnet-50 architecture. The highest accuracy was achieved with the Resnet-50 architecture, using Fine-tunning 50% training strategy.

Table 3
Comparison of deep learning architectures with other machine learning methods.

Approach	Training strategy	Training time (s)	Accuracy (%)
Inception-v3	FTuning25%	5,077.79	91.87
Resnet-50	FTuning50%	4,968.79	**93.82**
VGG-16	FTuning100%	4,884.36	91.80
VGG-19	FTuning50%	4,909.60	91.33
Xception	FTuning75%	5,330.41	90.52
SVM	Combined feature	43.93	**60.46**
Random Forest	extraction based on	9.83	56.42
J48	color[a], gredient[b],	5.57	48.28
Naive Bayes	texture[c] and shape [d]	0.25	12.80
k-NN		0.01	42.04
AdaBoost		0.47	47.18
SIFT	SVM e k = 25	7,730.66	48.80
SIFT	SVM e k = 50	12,101.26	51.40
SIFT	SVM e k = 100	18,710.90	**52.13**
SURF	SVM e k = 25	7,391.27	48.73
SURF	SVM e k = 50	13,238.70	49.53
SURF	SVM e k = 100	23,487.67	50.73
OPFSEMImst	($Z_1' = 10\%$ e $Z_1'' = 90\%$)	1.78	51.28
	($Z_1' = 20\%$ e $Z_1'' = 80\%$)	1.79	52.29
	($Z_1' = 30\%$ e $Z_1'' = 70\%$)	1.80	52.34
	($Z_1' = 40\%$ e $Z_1'' = 60\%$)	1.80	52.72
	($Z_1' = 50\%$ e $Z_1'' = 50\%$)	1.81	52.63
	($Z_1' = 60\%$ e $Z_1'' = 40\%$)	2.32	52.95
	($Z_1' = 70\%$ e $Z_1'' = 30\%$)	2.33	**53.61**
	($Z_1' = 80\%$ e $Z_1'' = 20\%$)	2.39	53.19
	($Z_1' = 90\%$ e $Z_1'' = 10\%$)	2.41	53.52

[a] (Swain and Ballard, 1991)
[b] (Dalal and Triggs, 2005)
[c] (Haralick, 1979; Ojala et al., 2002)
[d] (Hu, 1962)

6.3. Detection of soybean pests with UAV images

This section presents the approach proposed in Section 4 for the detection of soybean pests with UAV images. Fig. 11 shows the final step of our computer vision system by classifying the segments of an image of the plantation captured by the UAV at 2 m in height. Here, the

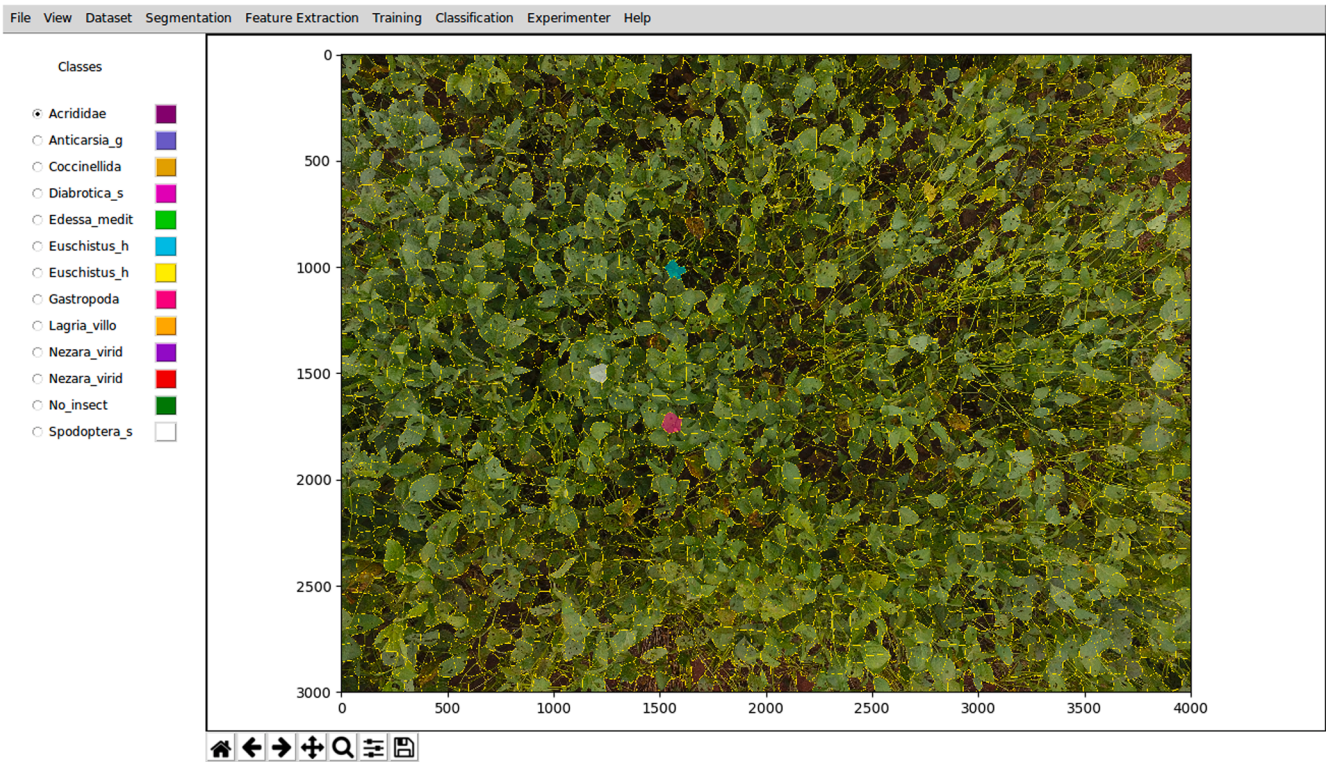

Fig. 11. Screenshot of our computer vision system detecting soybean pests with UAV images. The system called PYNOVISÃO presents the pests segmentation step and the superpixels classification using Resnet-50 architecture. Color labels stand the categories for our problem. PYNOVISÃO was registered by INPI under the number BR 51 2019 000427 2.

parameter $k = 2,000$ was adjusted to better segment the pests in the image.

According to Fig. 11, pests can be detected in the image by the color of the segment corresponding to their respective class. In the image, it is possible to observe the existence of three distinct segments of the others, that is, of those segments that do not have pests: 1 cyan segments, 1 magenta segment and 1 white segment, which correspond to the classes *Euschistus heros* adult, *Gastropoda* and *Spodoptera*spp., respectively. However, the pests belonging to the classes *Euschistus heros* nymph and *Lagria villosa* were not detected in the image by our software, even though their presence by the entomologist biologist was confirmed.

The set of training and test images of the system presented in Section 5 and statistically evaluated for performance in Section 6 was constructed with images collected by a smartphone camera at 50*cm* away from the target of interest (pest). In this section, we tested a UAV image captured at 2 m of height of the plantation, but we did not have good results for the accuracy in the classification task, assuming that, for higher altitudes, the size of the pests in the image gradually reduces and, consequently, the pest resolution in the image decreases, impacting the performance of the computer vision system. We do not statistically evaluate the performance captured images by UAV because they do not correspond to the same capture distance from our set of training and test images collected by the smartphone camera. As an alternative, high spatial resolution sensors (e.g., 100 or 200 megapixels) could be embarked in a UAV to capture high-definition images of pests in flight at higher altitudes, this demonstrates the potential of the proposed approach, although the investment of such sensors is still high.

7. Conclusion

In this paper, we evaluated the performance of the Inception-v3, Resnet-50, VGG-16, VGG-19 and Xception deep learning architectures for the classification of soybean pest images. An image segmentation step with the SLIC superpixels algorithm was considered to segment the pests in the images collected under real conditions. During the classification task, the performance of the deep learning architectures for different fine-tuning and transfer learning strategies was compared with other traditional feature extraction and learning approaches. Experimental results showed that deep learning architectures trained with fine-tuning computed weights lead to higher classification rates when compared with other machine learning methods, reaching accuracy of up to 93.82% with the Resnet-50 architecture. The results indicate that the architectures evaluated generalize well in soybean pests datasets and can support specialists and farmers in monitoring and controlling pests in soybean fields. As part of the future work, we intend to embark higher resolution cameras in the UAV and evaluate the performance of our approach to images collected at different heights. We also consider to evaluate the pest management by automatic counting the insects in the images captured by a UAV.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the National Center for Scientific and Technological Development (CNPq), the Coordination for the Improvement of Higher Education Personnel (CAPES) for the doctoral scholarship granted to three authors, NVIDIA Corporation for the donation of the graphic card and the Foundation to Support the Development of Teaching, Science and Technology of the state of Mato Grosso do Sul (FUNDECT) for the

financing of the project that originated this work.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S., 2012. Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. 34 (11), 2274–2282. https://doi.org/10.1109/TPAMI.2012.120.

Al-Saqer, S., Hassan, G.M., 2011. Artificial Neural Networks Based Red Palm Weevil (Rynchophorus Ferrugineous, Olivier) Recognition System. Am. J. Agric. Biol. Sci. 6, 356–364. https://doi.org/10.3844/ajabssp.2011.356.364.

Amorim, W.P., Falcão, A.X., Papa, J.P., Carvalho, M.H., 2016. Improving semi-supervised learning through optimum connectivity. Pattern Recogn. 60, 72–85. https://doi.org/10.1016/j.patcog.2016.04.020.

Amorim, W.P., Tetila, E.C., Pistori, H., Papa, J.P., 2019. Semi-supervised learning with convolutional neural networks for UAV images automatic recognition. Comput. Electron. Agric. 164 (2019), 104932. https://doi.org/10.1016/j.compag.2019.104932.

Barbedo, J.C.A., 2014. Using Digital Image Processing for Counting Whiteflies on Soybean Leaves. J. Asia-Pacific Entomol. 17 (4), 685–694. https://doi.org/10.1016/j.aspen.2014.06.014.

Bay, H., Ess, A., Tuytelaars, T., Gool, L.V., 2008. Speeded-Up Robust Features (SURF). Comput. Vis. Image Underst. 110 (3), 346–359. https://doi.org/10.1016/j.cviu.2007.09.014.

Brodbeck, C., Sikora, E., Delaney, D., Pate, G., Johnson, J., 2017. Using Unmanned Aircraft Systems for Early Detection of Soybean Diseases. Precision Agric. 8 (2), 802–806. https://doi.org/10.1017/S2040470017001315.

Calderón, R., Navas-Cortés, J.A., Zarco-Tejada, P.J., 2015. Early Detection and Quantification of Verticillium Wilt in Olive Using Hyperspectral and Thermal Imagery over Large Areas. Remote Sens. 7 (5), 5584–5610. https://doi.org/10.3390/rs70505584.

Chelladurai, V., Karuppiah, K., Jayas, D.S., Fields, P.G., White, N.D.G., 2014. Detection of Callosobruchus maculatus (F.) infestation in soybean using soft X-ray and NIR hyperspectral imaging techniques. J. Stored Prod. Res. 57, 43–48. https://doi.org/10.1016/j.jspr.2013.12.005.

Chollet F., 2015. Keras. [Online]. Available: https://github.com/fchollet/keras.

Chollet F., 2017. Xception: Deep Learning with Depthwise Separable Convolutions. 1800–1807. doi:10.1109/CVPR.2017.195.

Corrâa-Ferreira B.S. AMOSTRAGEM DE PRAGAS DA SOJA. In: Hoffmann-Campo C.B., Corrâa-Ferreira B.S., Moscardi F., 2012. Soja: manejo integrado de insetos e outros artrópodes-praga. Londrina: Embrapa Soja, cap 9, p. 631–672. ISBN 978-85-7035-139-5.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition (CVPR 2005). IEEE Comput. Soc. Conf. 1, 886–893. https://doi.org/10.1109/CVPR.2005.177.

da Silva, F.L., Sella, M.L.G., Francoy, T.M., Costa, A.H.R., 2015. Evaluating classification and feature selection techniques for honeybee subspecies identification using wing images. Comput. Electron. Agric. 114, 68–77. https://doi.org/10.1016/j.compag.2015.03.012. ISSN 0168–1699.

dos Santos, A.F., Freitas, D.M., Silva, G., Pistori, H., Folhes, M.T., 2017. Weed detection in soybean crops using ConvNets. Comput. Electron. Agric. 143, 314–324. https://doi.org/10.1016/j.compag.2017.10.027.

Fuentes, A., Yoon, S., Kim, S.C., Park, D.S., 2017. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. Sensors 17 (9), 2022. https://doi.org/10.3390/s17092022.

Garcia-Ruiz, F., Sankaran, S., Maja, J.M., Lee, W.S., Rasmussen, J., Ehsani, R., 2013. Comparison of two aerial imaging platforms for identification of huanglongbing-infected citrus trees. Comput. Electron. Agric. 91, 106–115. https://doi.org/10.1016/j.compag.2012.12.002.

Gedeon, C.I., Flórián, N., Liszli, P., Hambek-Oláh, B., Bánszegi, O., Schellenberger, J., Dombos, M., 2017. An Opto-Electronic Sensor for Detecting Soil Microarthropods and Estimating Their Size in Field Conditions. Sensors (Basel, Switzerland) 17 (8), 1757. https://doi.org/10.3390/s17081757.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14). IEEE Computer Society, Washington, DC, USA, pp. 580–587. https://doi.org/10.1109/CVPR.2014.81.

Guadarrama S., Silberman N., 2016. TF-Slim. [Online]. Available: https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/slim.

Guoguo, Y., Yidan, B., Ziyi, L., 2017. Localization and recognition of pests in tea plantation based on image saliency analysis and convolutional neural network. Editorial Office of Trans. Chin. Soc. Agric. Eng. 33 (6), 156–162. https://doi.org/10.11975/j.issn.1002-6819.2017.06.020.

Haralick, R.M., 1979. Statistical and structural approaches to texture. Proc. IEEE 67 (5), 786–804. https://doi.org/10.1109/PROC.1979.11328.

Hartigan, J.A., Wong, M.A., 2013. A k-means clustering algorithm. Appl. Stat. 28, 100–108 [Online]. Available: https://www.researchgate.net/publication/310403387_A_K-means_clustering_algorithm.

He K., Zhang X., Ren S., Sun J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: IEEE International Conference on Computer Vision (ICCV). IEEE Computer Society, Washington, DC, USA, 1026–1034. doi:10.1109/ICCV.2015.123.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., Kingsbury, B., 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The shared views of four research groups. IEEE Signal Process. Mag. 29 (6), 82–97. https://doi.org/10.1109/MSP.2012.2205597.

Hou, J., Wang, C., Hong, X., Zhao, C., Xue, C., Guo, N., Gai, J., Xing, H., 2011. Association analysis of vegetable soybean quality traits with SSR markers. Plant Breed 130 (4), 444–449. https://doi.org/10.1111/j.1439-0523.2011.01852.x.

Hu, M.-K., 1962. Visual Pattern Recognition by Moment Invariants. IRE Transaction of Information Theory IT-8. IRE Trans. Inform. Theory 8, 179–187. https://doi.org/10.1109/TIT.1962.1057692.

ImageNet, 2016. About ImageNet. ImageNet, 2016. [Online]. Available: http://www.image-net.org/about-overview.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, pp. 1725–1732. https://doi.org/10.1109/CVPR.2014.223.

Keyvan, A.V., Jafar, M., 2013. Performance evaluation of a machine vision system for insect pests identification of field crops using artificial neural networks. Arch. Phytopathol. Plant Protect. 46 (11), 1262–1269. https://doi.org/10.1080/03235408.2013.763620.

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

Leow, L.K., Chew, L.L., Chong, V.C., Dhillon, S.K., 2015. Automated identification of copepods using digital image processing and artificial neural network. BMC Bioinform. 16 (18), 1471–2105. https://doi.org/10.1186/1471-2105-16-S18-S4.

Liu, H., Lee, S.-H., Chahl, J.S., 2017a. A review of recent sensing technologies to detect invertebrates on crops. Precision Agric. 18 (4), 635–666. https://doi.org/10.1007/s11119-016-9473-6.

Liu, H., Lee, S.H., Chahl, J.S., 2017b. An evaluation of the contribution of ultraviolet in fused multispectral images for invertebrate detection on green leaves. Precision Agric. 18 (4), 667–683. https://doi.org/10.1007/s11119-016-9472-7.

Lowe, D.G., 1999. Object Recognition from Local Scale-Invariant Features. In: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2. IEEE Computer Society, Washington, DC, USA, pp. 1150–1157. https://doi.org/10.1109/ICCV.1999.790410.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965.

Lu, J., Ehsani, R., Shi, Y., de Castro, A.I., Wang, S., 2018. Detection of multi-tomato leaf diseases (late blight, target and bacterial spots) in different stages by using a spectral-based sensor. Scient. Rep. 8 (1), 2793. https://doi.org/10.1038/s41598-018-21191-6.

Machado, B.B., Orue, J.P.M., Arruda, M.S., Santos, C.V., Sarath, D.S., Goncalves, W.N., Silva, G.G., Pistori, H., Roel, A.R., Rodrigues-Jr, J.F., 2016. BioLeaf: A professional mobile application to measure foliar damage caused by insect herbivory. Comput. Electron. Agric. 129, 44–55. https://doi.org/10.1016/j.compag.2016.09.007.

Maharlooei, M., Sivarajan, S., Bajwa, S., Harmon, J., Nowatzki, J., 2017. Detection of soybean aphids in a greenhouse using an image processing technique. Comput. Electron. Agric. 132, 63–70. https://doi.org/10.1016/j.compag.2016.11.019.

Mahlein, A.-K., Oerke, E.-C., Steiner, U., Dehne, H.-W., 2012. Recent advances in sensing plant diseases for precision crop protection. Eur. J. Plant Pathol. 133 (1), 197–209. https://doi.org/10.1007/s10658-011-9878-z.

Martineau, M., Conte, D., Raveaux, R., Arnault, I., Munier, D., Venturini, G., 2017. A survey on image-based insects classification. Pattern Recogn. 65, 273–284. https://doi.org/10.1016/j.patcog.2016.12.020.

Oerke, E.-C., Steiner, U., Dehne, H.-W., Lindenthal, M., 2006. Thermal imaging of cucumber leaves affected by downy mildew and environmental conditions. J. Exp. Bot. 57 (9), 2121–2132. https://doi.org/10.1093/jxb/erj170.

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 971–987. https://doi.org/10.1109/TPAMI.2002.1017623.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191.

Pantazi, X.E., Tamouridou, A.A., Alexandridis, T.K., Lagopodi, A.L., Kashefi, J., Moshou, D., 2017. Evaluation of hierarchical self-organising maps for weed mapping using UAS multispectral imagery. Comput. Electron. Agric. 139 (2017), 224–230. https://doi.org/10.1016/j.compag.2017.05.026.

Peruca, R.D., Coelho, R.G., da Silva, G.G., Pistori, H., Ravaglia, L.M., Roel, A.R., Alcantara, G.B., 2018. Impacts of soybean-induced defenses on Spodoptera frugiperda (Lepidoptera: Noctuidae) development. Arthropod-Plant Interact. 12 (2), 257–266. https://doi.org/10.1007/s11829-017-9565-x.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vision (IJCV) 115 (3), 211–252 arXiv:1409.0575.

Shajahan S., Sivarajan S., Maharlooei M., Bajwa S., Harmon J., Nowatzki J., Igathinathane C., 2016. Identification and Counting of Soybean Aphids from Digital Images using Particle Separation and Shape Classification. Conference: ASABE Annual International Meeting, At Orlando, Florida. doi:10.13031/aim.20162462927.

Simonyan K., Zisserman A., 2014. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations (ICLR2015). arXiv:1409.1556.

Sirisomboon, P., Hashimoto, Y., Tanaka, M., 2009. Study on non-destructive evaluation methods for defect pods for green soybean processing by near-infrared spectroscopy. J. Food Eng. 93 (4), 502–512. https://doi.org/10.1016/j.jfoodeng.2009.02.019.

Swain, M.J., Ballard, D.H., 1991. Color indexing. Int. J. Comput. Vision 7 (1), 11–32. https://doi.org/10.1007/BF00130487.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Comput. Vision Pattern Recogn. (CVPR). https://doi.org/10.1109/CVPR.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 arXiv:1512.00567.

Tetila E.C., 2018. INSection 5K13C - Image dataset of soybean pests. Available: https://bit.ly/2SKp9jC.

Tetila, E.C., Machado, B.B., Belete, N.A.S., Guimarães, D.A., Pistori, H., 2017. Identification of Soybean Foliar Diseases Using Unmanned Aerial Vehicle Images. IEEE Geosci. Remote Sens. Soc. 14 (12), 2190–2194. https://doi.org/10.1109/LGRS.2017.2743715.

Tetila, E.C., Machado, B.B., Menezes, G.K., Oliveira-Jr, A.S., Alvarez, M., Amorim, W.P., Belete, N.A.S., Silva, G.G., Pistori, H., 2019a. Automatic Recognition of Soybean Leaf Diseases Using UAV Images and Deep Convolutional Neural Networks. IEEE Geosci. Remote Sens. Lett. https://doi.org/10.1109/LGRS.2019.2932385.

Tetila, E.C., Machado, B.B., Menezes, G.V., Belete, N.A.S., Astolfi, G., Pistori, H., 2019b. A Deep-Learning Approach for Automatic Counting of Soybean Insect Pests. IEEE Geosci. Remote Sens. Lett. https://doi.org/10.1109/LGRS.2019.2954735.

Yaakob, S.N., Jain, L., 2012. An insect classification analysis based on shape features using quality threshold ARTMAP and moment invariant. Appl. Intell. 37 (1), 12–30. https://doi.org/10.1007/s10489-011-0310-3.

Yanan, M., Min, H., Bao, Y., Qibing, Z., 2014. Automatic threshold method and optimal wavelength selection for insect-damaged vegetable soybean detection using hyperspectral images. Comput. Electron. Agric. 106, 102–110. https://doi.org/10.1016/j.compag.2014.05.014.

Wang N., Yeung D.-Y., 2013. Learning a deep compact image representation for visual tracking. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS), USA, 1:809–817. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999611.2999702.

Wang, J., Lin, C., Ji, L., Liang, A., 2012. A new automatic identification system of insect images at the order level. Know.-Based Syst. 33, 102–110. https://doi.org/10.1016/j.knosys.2012.03.014.

Weiss U., Biber P., Laible S., Bohlmann K., Zell A., 2010. Plant Species Classification Using a 3D LIDAR Sensor and Machine Learning. Ninth International Conference on Machine Learning and Applications, Washington, DC, 339-345. doi:10.1109/ICMLA.2010.57.

Wen, C., Wu, D., Hu, H., Pan, W., 2015. Pose estimation-dependent identification method for field moth images using deep learning architecture. Biosyst. Eng. 136, 117–128. https://doi.org/10.1016/j.biosystemseng.2015.06.002.